LEVEL

DTNSRDC/SPD-0867-01

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Md. 20084

ADA078605

A NONLINEAR MATHEMATICAL MODEL OF MOTIONS OF A PLANING BOAT IN IRREGULAR WAVES

by

ERNEST E. ZARNICK

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

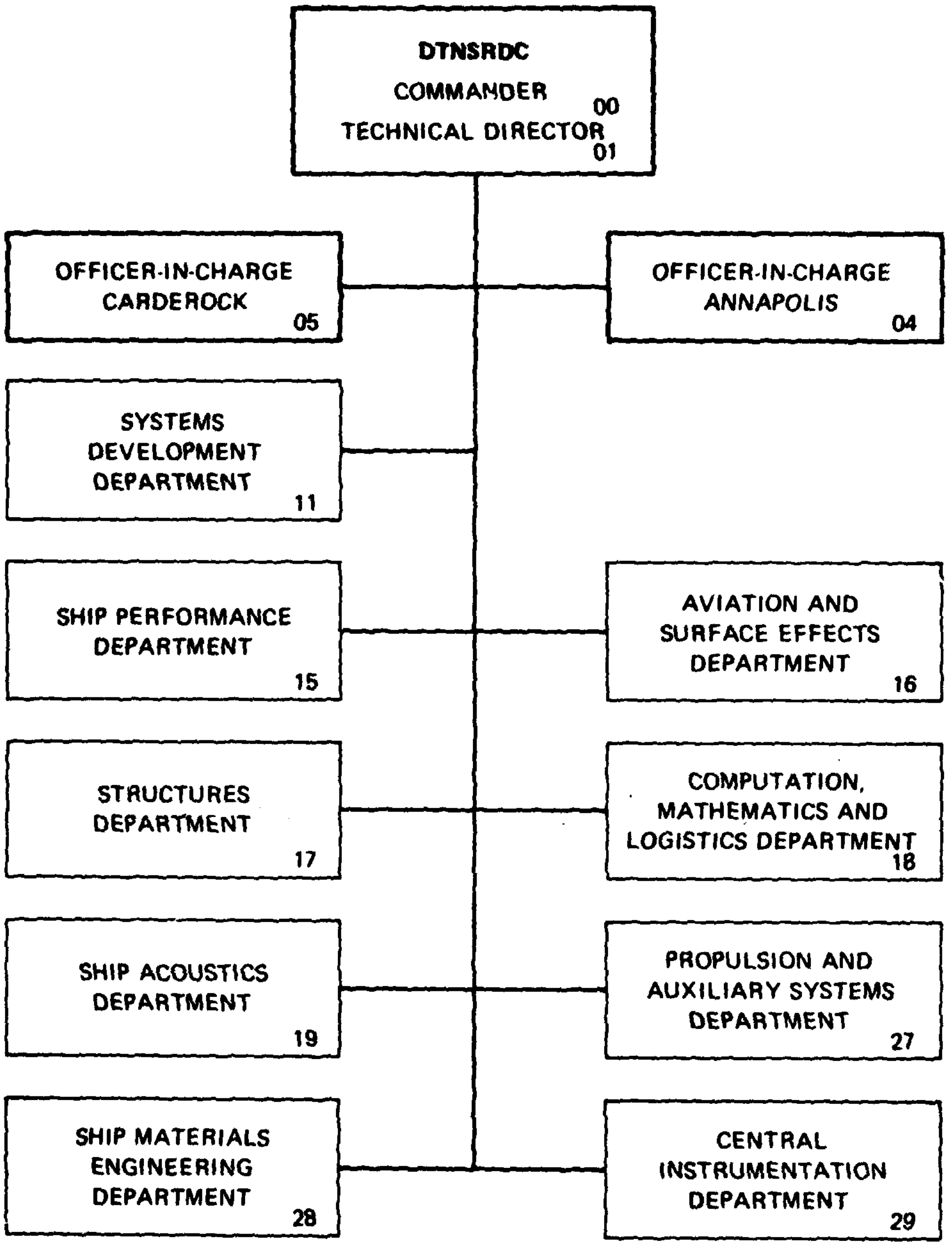
SHIP PERFORMANCE DEPARTMENT

SEPTEMBER 1979

DTNSRDC/SPD-0867-01

A NONLINEAR MATHEMATICAL MODEL OF MOTIONS OF A PLANING BOAT IN IRREGULAR WAVES

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) DTNSRDC/SPD-0867-01	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) A NONLINEAR MATHEMATICAL MODEL OF MOTIONS OF A PLANING BOAT IN IRREGULAR WAVES		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Ernest E. Zarnick		8. CONTRACT OR GRANT NUMBER(s) (12) 61
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship R&D Center Bethesda, Maryland		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element 625 43 N Task Area ZF43-421-001 Work Unit No. 1500-102
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Material Command (08T2) Washington, DC 20362		12. REPORT DATE (11) Sep[redacted]1979
		13. NUMBER OF PAGES 55
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (16) F43421 (17) ZF43421...		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Planing Boat, Hydrodynamic Impact, Small Boat Seaworthiness, Nonlinear Ship Motions in Waves

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A computer program previously developed to estimate the motions and accelerations of a planing craft in regular waves was modified and extended to compute the corresponding motions in random or irregular waves. Ten regular waves with random phase were combined to represent the random seaway. The amplitudes and frequencies that were selected represent the energy distribution of a Pierson-Moskowitz spectrum for a fully developed sea. A comparison of computed results with experiment indicates that the computer program can predict craft behavior with reasonable quantitative

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

374694

accuracy in moderate operating conditions. In severe operating conditions, however, the amplitudes of the computed vertical accelerations, which include impacts, are one-half of the experimental values.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATION

A_R	Section Area
a	Correction factor for buoyancy force
b	Half-beam of craft
c_i	Wave celerity of ith component
$C_{D,c}$	Crossflow drag coefficient
CG	Center of Gravity of boat
C_Δ	Load coefficient $\Delta/\rho g(2b)^3$
C_λ	Wavelength coefficient $L/\lambda \; C_\Delta/(L/2b)^{2\ 1/3}$
D	Friction drag force
F_x	Total hydrodynamic force in x direction
F_θ	Total hydrodynamic moment about pitch axis
f	Two-dimensional hydrodynamic force
g,G	Acceleration of gravity
H	Wave height, crest to trough
h	Vertical submergence of point below free surface
h_{50}	Heave crest or trough value corresponding to 50% probability point
h_{90}	Heave crest or trough value corresponse to 90% probability point
I	Pitch moment of inertia
I_a	Added pitch, moment of inertia
k_a	Two-dimensioal added-mass coefficient
k_i	Wave number of ith component
L	Hull length
LCG	Longitudinal center of gravity percent of L
M	Mass of craft

NOTATION (CONT)

M_a	Added mass of craft
m_a	Sectional (two-dimensional) added mass
N	Hydrodynamic force normal to baseline
r'	Ratio of negative maximum to total maximum
r	Wave elevation, positive down, feet
r_i	Wave amplitude of ith component
U	Relative fluid velocity parallel to baseline
V	Relative fluid velocity normal to baseline
$V/\sqrt{L}$	Speed-to-length ratio in knots/ft$^{1/2}$
W	Weight of craft
w_z	Vertical component of wave orbital velocity, positive down
x	Fixed longitudinal axis; also the coordinate of a point relative to the origin of body axis.
$\dot{x}_{CG}$	Surge velocity
$\ddot{x}_{CG}$	Surge acceleration
x_{CG}	Longitudinal distance from origin of fixed axis to CG of the body
z	Fixed vertical normal axis; also the coordinate of a point relative to the origin of body axis, positive down
$\ddot{z}_{CG}$	Heave acceleration of the CG
z_{CG}	Heave displacement of the CG, positive down
β	Deadrise angle
Δ	Hull displacement W
ζ	Normal body axis; also the coordinate of a point relative to the origin of body axis
η	Vertical acceleration (i.e., in direction of ζ axis)

NOTATION (CONT)

θ	Pitch angle
$\dot{\theta}$	Pitch angular velocity
$\ddot{\theta}$	Pitch angular acceleration
θ_{50}	Pitch crest or trough corresponding to 50% probability point, degrees
θ_{90}	Pitch crest or trough corresponding to 90% probability point, degrees
ξ	Longitudinal body axis; also the coordinate of a point relative to the origin of body axis
ρ	Density of water
σ_i	Phase angle of ith component
ω	Wave frequency
ω_p	Peak frequency of wave spectrum
ℓ	Wetted length
Ω	Nondimensional frequency, ω/ω_p
ε	Spectral width parameter

ABSTRACT

A computer program previously developed to estimate the motions and accelerations of a planing craft in regular waves was modified and extended to compute the corresponding motions in random or irregular waves. Ten regular waves with random phase were combined to represent the random seaway. The amplitudes and frequencies that were selected represent the energy distribution of a Pierson-Moskowitz spectrum for a fully developed sea. A comparison of computed results with experiments indicate that the computer program can predict craft behavior with reasonable quantitative accuracy in moderate operating conditions. In severe operating conditions, however, the amplitudes of the computed vertical accelerations, which include impacts. are one-half of the experimental value.

ADMINISTRATIVE INFORMATION

This work has been authorized by the Naval Material Command (08T2); under Program Element 625 43 N, Task Area ZF43-421-001, administered by the Ship Performance Department, High Performance Vehicle Program Office, Code 1512.

INTRODUCTION

In a previous study[1] a computer program was developed to estimate the motions and accelerations of a planing craft in regular waves. As a logical extension of this work, the program was modified to compute the motion of the craft in random or irregular waves.

Since the mathematical model is nonlinear, the computations are made in the time domain. Ten regular waves are combined with random phase to represent the random seaway. The amplitudes and frequencies are adjusted to conform to the energy distribution in a Pierson-Moskowitz fully developed sea.

The mathematical model was developed for a V-shaped prismatic-body with hard chines and constant deadrise planing at constant speed. The thrust and the friction drag forces are assumed to act through the center of gravity. The vertical components of the thrust and fiction drag are also assumed to be negligible in comparison to the hydrodynamic forces.

The mathematical formulation is analogous to low-aspect-ratio wing theory with provisions for including hydrodynamic impact loads, essentially a strip theory. Surface wave generation and forces associated with unsteady circulatory flow are neglected, and the flow is treated as quasi-steady. The mathematical formulation is an empirical synthesis of several theoretically derived flows describing the overall craft hydrodynamics.

MATHEMATICAL FORMULATION

EQUATIONS OF MOTIONS

The equations of motion for a planing craft restricted to pitch θ, heave z_{CG}, and surge x_{CG} can be written as

$$M\ddot{x}_{CG} = T_x - N \sin\theta - D \cos\theta$$
$$M\ddot{z}_{CG} = T_z - N \cos\theta + D \sin\theta + W$$
$$I\ddot{\theta} = Nx_c - Dx_d + Tx_p$$

where M is mass of craft

- I is pitch moment of inertia of craft
- N is hydrodynamic normal force
- D is friction drag
- W is weight of craft
- T_x is thrust component in x direction
- x_c is distance from center of gravity (CG) to center of pressure for normal force, positive forward
- x_d is distance from CG to line of action for friction drag force
- x_p is moment arm of thrust about CG.

Motions are measured relative to a fixed coordinate system with the x axis located in the undisturbed free surface pointing in the direction of travel and the z axis pointing downward.

Since the perturbation velocities in the forward direction are small in comparison to the speed of the craft, the equations of motion can be simplified by neglecting them and by setting the forward velocity equal to a constant, i.e.,

$$\dot{X}_{CG} = \text{CONSTANT}$$

Furthermore, if it is assumed that the vertical components of the thrust and friction drag forces are small in comparison to the hydrodynamic forces

and that the total thrust and friction drag forces are acting through the center of gravity (so as to produce no moments) the equations of motion can be written as

$$\ddot{x}_{CG} = 0$$

$$M\ddot{z}_{CG} = -N\cos\theta + W$$

$$I\ddot{\theta} = Nx_c$$

A so called "strip theory" is used to obtain the hydrodynamic force acting on the body by integrating the 2-D hydrodynamic forces normal to the baseline over the wetted length of the body. A body coordinate system (ξ,ζ) with its origin at the CG and the ξ axis pointing forward parallel to the baseline of the body as shown in Figure 1 is used to facilitate this integration.

The normal hydrodynamic force per unit length f, acting at a section, is assumed to be proportional to the rate of change of momentum associated with an added mass term and the cross flow drag, i.e.

$$f = \frac{D}{Dt}(m_a V) + C_{D,c}\rho b V^2$$

where V is the velocity in plane of the cross section normal to the baseline

m_a is the added mass associate with the section form

$C_{D,c}$ is the crossflow drag coefficient

ρ is the density of the fluid

b is the half beam

Expanding the momentum term results is

$$\frac{D}{Dt}(m_a V) = m_a\dot{V} + V\dot{m}_a - \frac{\partial}{\partial\xi}(m_a V)\frac{d\xi}{dt}$$

where ξ is the body coordinate parallel to the baseline; see Figure 1.

The last term on the right-hand side of the above equation takes into account the variation of the section added mass along the hull. This contribution can be visualized by considering the 2-D flow plane as a substantive surface moving past the body with velocity $U = -d\xi/dt$ tangent to the baseline. As the surface moves past the body, the section geometry in the moving surface may change with a resultant change in added mass. This term exists evan in steady-state conditions and is the lift-producing factor in low-aspect-ration theory.

For a V-shaped wedge the 2-D added mass is defined as

$$m_a = k_a \frac{\pi}{2}\rho\ b^2\ (\xi,\zeta,t)$$

where k_a is the added mass coefficient (assumed to be 1 in this study) and b is the wetted half beam. Once the chine becomes wetted the beam is assumed constant regardless of depth of penetration.

Cross-flow drag for a V section with separation at the chine is assumed equal to the drag of a flat plate ($C_{D,c}$ = 1.0) corrected by the Bobyleff flow coefficient approximated by cos β, i.e.,

$$C_{D,c} = 1.0 \cos \beta$$

The Bobyleff flow coefficient is the theoretical ratio of the pressure on a V-section to that experienced by a flat plate for Helmholtz-type flow.

The same approximation is used for estimating the drag coefficient for nonwetted chine sections, using the instantaneous value of the half-beam at the free surface.

An additional force acting on the body is the buoyancy force f_B. This force is assumed here in to act in the vertical direction and to be equal to the static buoyancy force multiplied by a correction factor,

i.e.,

$$f_B = a\rho g A_R$$

where A_R is the cross-sectional area of the section, and a is a correction factor. The full amount of the static buoyancy is not realized because at planing speeds the water separates from the transom and chines, reducing the pressure at these locations to atmospheric or less than the equivalent hydrostatic pressure. A greater reduction is realized in the buoyancy moment because of the corresponding shift in the center of pressure. Shuford[2] in his work on steady-state planing recommended a factor of one-half to obtain the correct buoyancy force. In the following computations, the buoyancy force was corrected by a factor of one-half, i.e., a = 1/2. The buoyancy moments, computed as the static buoyancy force multiplied by its corresponding moment arm, was corrected by an additional factor of one-half to obtain the proper mean-trim angles.

Integrating the 2-D hydrodynamic force over the wetted length of the craft (ℓ) and taking the component in the z direction results in

$$\begin{aligned} -N\cos\theta = F_z(t) &= \int_\ell f\cos\theta\, d\xi + \int_\ell f_B\, d\xi \\ &= -\Bigg[\int_\ell \Big\{ m_a(\xi,t)\dot{V}(\xi,t) + \dot{m}_a(\xi,t)V(\xi,t) \\ &\quad - U(\xi,t)\frac{\partial}{\partial\xi}[m_a(\xi,t)V(\xi,t)] \\ &\quad + C_{D,c}(\xi,t)\rho b(\xi,t)V^2(\xi,t)\Big\}\cos\theta\, d\xi \\ &\quad + a\rho g A_R d\xi\Bigg] \end{aligned}$$

Similarly, the hydrodynamic moment about the CG is obtained by integrating the product of the normal force per unit length by the corresponding moment arm.

$$F_\theta = -\int_\ell f(\xi,t)\xi d\xi - \int_\ell f_b \cos\theta\, \xi d\xi$$

$$= \int_\ell \Big\{ m_a(\xi,t)\dot{V}(\xi,t) + \dot{m}_a(\xi,t)V(\xi,t)$$

$$- U(\xi,t)\frac{\partial}{\partial\xi}(m_a(\xi,t)V(\xi,t)) + C_{D,c}(\xi,t)\rho b(\xi,t)V^2(\xi,t)$$

$$+ a\frac{\rho}{2}gA_R\cos\theta \Big\} \xi d\xi$$

Wave excitation enters into the above equations through the geometrical properties of the wave, altering the wetted length and draft of the craft, and by the vertical component of the wave orbital velocity at the surface w_z, altering the normal velocity V. Diffraction has been neglected.

The horizontal component of orbital veolcity is neglected, since it is assumed small in comparison with the forward speed $\dot{x}_{CG}$. The velocities U and V may then be written as

$$U = \dot{x}_{CG}\cos\theta - (\dot{z}_{CG} - w_z)\sin\theta$$
$$V = \dot{x}_{CG}\sin\theta - \dot{\theta}\xi + (\dot{z}_{CG} - w_z)\cos\theta$$

The depth of submergence h of the body at any point $P(\xi,\zeta)$ may be determined by

$$h = z_{CG} - \xi\sin\theta + \zeta\cos\theta - r$$

where r is the instataneous value of the wave elevation directly above the point measured positive downward.

A more detailed derivation of the above integrals for the hydrodynamic force and moment is presented in reference (1). Although the hydrodynamic

forces and moment require integration over the wetted length, which may vary with time, the resulting equations of motion can be integrated in the time domain using numerical method such as the Runge-Kutta Merson integration routine used in these studies.

REPRESENTATION OF THE SEAWAY

The seaway in general can be represented by an infinite sum of sine waves with random phase. In these studies, for the sake of computational economy, the seaway is represented by the discrete sum of ten harmonic waves with random phase

$$r = \sum_{i=1}^{10} r_i \cos\left[k_i \ (x + c_i t) + \sigma_i\right]$$

where r_i is the wave amplitude

k_i is the wave number

c_i is the wave celerity

and σ_i is the random phase angle of the ith wave component.

Note that at point $P(\xi,\zeta)$ on the craft

$$x = x_{CG} + \xi \cos\theta + \zeta \sin\theta$$

and $$x_{CG} = \int \dot{x}_{CG} dt$$

Each frequency and wave amplitude is weighted in accordance with the energy distribution in a Pierson-Moskowitz spectrum for a fully developed sea. The Pierson-Moskowitz formulation for a continuous spectrum can be written as

$$S(w) = \frac{Ag^2}{\omega^5} \quad e^{-B/\omega^4}$$

where $A = 8.1 \times 10^{-3}$

g = gravitational constant

$$B = \frac{4Ag^2}{(H\ 1/3)^2}$$

The constant B is also related to the peak frequency (ω_p) of the spectrum

by

$$B = \frac{5}{4}\,\omega_p^4$$

which can be confirmed by differentiating the spectrum formulation and setting it equal to zero. Normalizing the frequencies by the peak frequency leads to a nondimensional spectrum $\overline{S}$ which is related to the dimensional spectrum by

$$\overline{S}(\Omega) = \frac{16\omega_p}{H_{1/3}^2}\, S(\omega{=}\Omega\omega_p) = \frac{5}{\Omega^5}\, e^{-5/4\Omega^4}$$

where $\Omega = \omega/\omega_p$

and where

$$\int_0^\infty \overline{S}\,(\Omega)d\Omega = 1$$

The discrete frequencies representing the spectrum varied from Ω = .80 to 2.6 in nearly equal increments $\Delta\Omega$=0.2. A slight random perturbation is given to each frequency to avoid precise integer multiple frequencies,thereby increasing the fundamental repetition period of the computed time history. Each discrete amplitude is adjusted so that its energy corresponds to that contained in a band width ($\Delta\Omega$) centered about its frequency in the continuous spectrum i.e.,

$$r_i^2 = \frac{H_{1/3}^2}{8} \cdot \int_{\Omega_i-\Delta\Omega/2}^{\Omega_i+\Delta\Omega/2} \overline{S}(\Omega)d\Omega$$

The band widths are equally spaced between frequencies except for the first and last frequencies which lump all of the remaining energy at the beginning and end of the spectrum. Table 1 presents a list of the amplitude for each nondimensional frequency in terms of the significant wave height.

COMPARISON OF COMPUTED RESULTS WITH EXPERIMENTS

Computations of pitch and heave motions and bow and CG vertical

accelerations were made using the computer program (see Appendix) for comparison with the model experiments of Fridsma[3]. Fridsma tested a series of constant-deadrise models of various lengths in irregular waves to determine the effects of deadrise, trim, load, speed and sea state on the added resistance, heave and pitch motions and vertical accelerations at the bow and CG. Figure 2 shows the lines of the prismatic models. The computations were made with the Centers' Control Data Corporation 6700 computer system. A listing of the computer program is presented in the Appendix.

Table 2 presents the characteristics of the model craft for those conditions selected for the comparison. The number of computer runs was kept to a minimum for economic reasons. Approximately one minute of central processor time was required for every second of data using model scale dimensions, and approximately 40 seconds of model scale data was required to obtain 100 cycles of amplitude data.

The output of the program is the time histories of the pitch and heave motions and the bow and CG accelerations. Sample plots of the outputs are shown in Figures 3 and 4. Procedures required for processing and analysing the data are not a part of this study. In order to facilitate the comparison, the analysis procedures followed in this report are those used by Fridsma[3] for his experimental data. The amplitudes (maxima or minima) of the pitch and heave motions about the mean are assumed to be described by the so called "Generalized Rayleigh Distribution." i.e.,

$$p(y) = \frac{1}{\sqrt{2\pi}} \quad \varepsilon e^{-1/2y^2/\varepsilon^2} + (1 - \varepsilon^2)^{1/2} y \; e^{-1/2y^2} \int_{-\infty}^{y(1-\varepsilon^2)\varepsilon} e^{-1/2x^2} dx$$

where y = maximum or minimum (absolute values) normalized by the standard deviation,

$$\varepsilon^2 = 1 - (1 - 2r')^2$$

and r' = ratio of negative maxima to total maxima or positive minima to total minima.

To fit the data to this distribution the crests (maxima) and troughs (minima) relative to the mean value are first determined from time histories of the motions. The mean value is defined as halfway between the average crest and average trough value. The crests or trough data (X_i) are sorted in ascending order and grouped into fifteen intervals. At the same time, the proportion r' of negative maxima to total maxima or positive minima to total minima is determined. The cumulative frequency and corresponding probability that a crest or trough is less than or equal to the interval value (X_i) is then computed. From the probability and r' values, the theoretical value of the normalized amplitude (y) is calculated. A plot of X_i versus corresponding y values is compared with a line drawn through x = y = 0, and the point, $x = \bar{x}$, $y = \bar{y} = \sqrt{\pi/2}(1-2r')$ which is indicative of the fit of the theoretical distribution function to the data. The values x and y are the observed average value of the first moment and the theoretical average value (normalized) respectively.

Figures 5 and 6 show typical examples of such plots for the pitch and heave crests and troughs. As can be seen in these figures, the data fit the assumed probability function reasonably well, but it is also quite possible that some other distribution might fit the data better.

The acceleration data were assumed to follow a simple exponential distribution. For this distribution, the probability, P of the acceleration peak η being less than a given value is

$$P(\eta) = 1 - e^{-\eta/\bar{\eta}}$$

where $\bar{\eta}$ = average peak acceleration.

Only the negative peak accelerations (impact spikes as well as wave induced) were analyzed. The data were sorted and grouped into fifteen intervals similar to the motion data analysis and the probability was plotted with respect to η on inverted semilog paper. For a good fit, the data should follow a straight line through the point ($P = 0.368$, $\eta = \bar{\eta}$) and the origin. Figure 7 shows a sample of the acceleration data plotted in the above manner. An exponential probability function appears to be a good fit to the data.

Table 3 presents a comparison of the computed motions and accelerations with the corresponding experimental results. The computations were made for a craft with a length to beam ration (L/b) of 5, a deadrise angle (β) of 20 degrees and a speed length ratio ($V/\sqrt{L}$) of 6 for several sea states with significant wave height to beam ratios ($H_{1/3}/2b$) of .222, .444 and .666 which would correspond to Sea states 2, 3, and 4 respectively for a 50 foot craft. Computations were also made in sea condition with $H_{1/3}/2b$ of .444 for a speed length ratio of 4 at 10, 20 and 30 degrees deadrise angles.

The tabulated values for the heave and pitch are those with a 50 percent and those with a 90 percent probability of not being exceeded. Heave is nondimensionalized by the beam. The values for the bow and CG accelerations are the average values of the negative peaks.

Other statistical variables such as the 1/3 or 1/10 highest values can be computed from the specific probability distribution. For the assumed distribution of the motion amplitudes, the 1/10 highest value is related to the 90% probability value by the ratio of $y_{1/10}/y_{90}$, which is

approximately 1.33 over the r' value range measured. The 1/10 highest accelerations from the exponential distribution is 3.30 $\bar{\eta}$.

Plots of the data in Table 3 are also presented in Figure 8 through 13. Figures 8 through 10 show respectively a comparison between the computed and experimental results of the variation in the 90 percent probability values for the pitch and heave motions, and the average values for the bow and CG accelerations, with significant wave height to beam ratio. Figures 11 through 13 show similar plots for the variation with deadrise angle.

The pitch data in the figures show that while the computed troughs (bow down) are in reasonable agreement with the experiment, the computer crests are lower than the experiment. The heave exhibits the same trend with the computed crests (CG down) being in reasonable agreement with the expeirment and troughs being lower than the experiment. Furthermore, the 90 percent probability values for the pitch and heave crests and troughs for the computer model are about equal in magnitude; whereas, the experimental model values in the pitch bow up and heave CG up direction are greater. It appears that the experiment model exhibits more nonlinearity than the computer model.

The computed acceleration data for the bow and CG are generally lower than the corersponding experiment data. Figures 10 and 13 show that the computed accelerations differ by 15 percent to 50 percent of the corresponding experiment values with the largest differences occuring in the more severe conditions where the accelerations are extremely high.

For example, at the ten degree deadrise angle condition presented in Table 2, the average 1/10 highest bow acceleration corresponds to about

24 g's for the experiment and 11 g's for the computed value. It is doubtful that any operational boat would be driven to such extreme conditions.

The characteristics of the experiment model motions appear to be slightly different from those of the computer model in very extreme conditions. The experiment model experienced larger pitch bow up and heave motion than the computer model, which probably resulted in larger impact accelerations. This does not completely explain the differences between the experiment and computed vertical accelerations which may also reflect a deficiency in the mathematical representation of the impact phenomenon, and perhaps the seaway as well.

In less severe conditions the agreement between computed and experiment results is better. Good results can be expected for speed length ratios up to approximately 6 in a seaway with significant wave height to beam ratio of 0.222 (State 2 sea for a 50 foot craft). For a significant wave height to beam ratio of 0.444 (State 3 for a 50 foot craft), the calculations could probably be used to predict reasonable quantitative results up to a speed length ratio of 4. In more extreme conditions the computed results are less accurate quantitatively, but are still indicative of gross trends.

SUMMARY AND CONCLUSIONS

A computer program was developed to compute the motions and accelerations of simple prismatic planing craft in head irregular waves. This was achieved by incorporating irregular waves into an existing program for computing the motions in regular waves. The irregular waves were synthesized by combining ten regular waves with random phase and with frequencies and amplitudes weighted to represent a Pierson-Moskowitz spectrum for fully developed seas.

Computations were made for a craft with a length to beam ratio of 5, a deadrise angle of 20 degrees and a speed length ratio of 6 for several sea states, and in a single sea state for a speed length ratio of 4 with 10, 20 and 30 degrees deadrise angles. The results were compared with the experiment results of Fridsma. First the probability distributions of the crests and troughs of the motions and acceleraions were examined to determine whether or not they were the same as those obtained by the experiments. It was found that the fit of the pitch and heave crests and troughs to the "Generalized Rayleigh Distribution" which was used in the experiment data analysis, was acceptable for the computed data, but no attempt was made to fit the data to other types of distribution which might have fitted better. The computed acceleration data, fitted an exponential type of distribution.

A comparison of the motions showed that the computed pitch troughs were in good agreement with the experiment, but the crests (bow up) were lower than the experiment values. The heave exhibited the same trend, with the troughs (CG heave up) for the experiment being higher than the computed values while the crests were in good agreement.

Computed vertical accelerations at the bow and CG were, for certain conditions, much lower than the experiment values. At some conditions examined, the computed accelerations were about half of the comparable experiment values. This occurred at the more extreme operating condition where very large acceleations were experienced. For example, a value of 24 g's was obtained for the average of the 1/10 highest bow acceleration from the experiment for a craft with 10 degrees deadrise angle as compared to 11 g's from the computed results. These values were obtained in a seaway

with a significant wave height to beam ratio of .666 and a speed length ratio of 6 which represent an operating condition far more severe than that in which the boat would be expected to operate (42 knots in a State 4 sea for a 50 foot craft).

In summary it appears that the computer program can predict craft behavior quantitatively most effectively in moderate operating conditions. In severe operating conditions, probably beyond that in which the craft would be expected to operate, the computed vertical accelerations which include impacts are roughly one half the experiment values; however, despite the deficiency of the computer program in predicting quantitative results, the predictions are still indicative of gross trends.

Additional work is required to improve the prediction of the impact accelerations especially during the more severe conditions. Towards this end, it would be desirable to compare the time histories of experimental motion and acceleration data with the corresponding computed values. It may be possible to modify the hydrodynamic coefficients in the mathematical model on the basis of experimental results, specifically those affecting impact acceleration, and greatly improve the correlation. It is recommended that if additional model experiments are conducted, time histories of the motions and accelerations along with the wave height be made available for the above studies.

ACKNOWLEDGMENT

Acknowledgment is given to Ms. Dana Gentily of ORI, Inc., who prepared the comptuer programs for statistical analysis of the data.

REFERENCES

1. Zarnick, E.E., "A Nonlinear Mathematical Model of a Planing Boat in Regular Waves," DTNSRDC Report 78/032 (March 1978).

2. Shuford, S.L.,Jr."A Theoretical and Experimental Study of Planing Surfaces Including Effects of Cross Section and Plan Form," National Advisory Committee for Aeronautics Report 1355 (1957).

3. Fridsma, G., "A Systematic Study of the Rough Water Performance of Planing Boats (Irregular Waves - Part II)," Davidson Laboratory Stevens Institute of Technology Report R1495 (March 1971).

TABLE 1

WAVE AMPLITUDE REPRESENTING

DISCRETE SPECTRUM

Ω	$r/H_{1/3}$
0.795	0.1364
1.0	0.1861
1.183	0.1657
1.403	0.1302
1.602	0.0999
1.795	0.0771
2.004	0.0604
2.194	0.0482
2.392	0.0390
2.612	0.0626

TABLE 2

MODEL CHARACTERISTICS AND WAVE

CONDITIONS FOR COMPUTATIONS

Model Length = 114.3 cm (3.75 ft)
Length/Beam = 5; C_{Δ} = 0.600

COMPUTER RUN	SYMBOL	β DEGREE	LCG PERCENT L	RADIUS OF GYRATION PERCENT L	$V/\sqrt{L}$	$H_{1/3}/2b$
1	M	20	64.0	24.8	6	0.222
2	M	20	64.0	24.8	6	0.444
3	M	20	64.0	24.8	6	0.666
4	O	20	66.8	25.0	4	0.666
5	C	10	68.0	25.0	6	0.444
6	G	30	62.1	25.0	6	0.444

TABLE 3 - COMPARISON OF COMPUTED RESULTS WITH EXPERIMENT

	PITCH (DEGREES)						HEAVE						ACCELERATION (Gs)	
	Crest			Trough			Crest			Trough				
	r	θ_{50}	θ_{90}	r	θ_{50}	θ_{90}	r	$h_{50}/2b$	$h_{90}/2b$	r	$h_{50}/2b$	$h_{90}/2b$	$\bar{n}_{bow}$	$\bar{n}_{CG}$
Computed	0.169	0.69	1.67	0.108	0.82	1.75	0.122	0.027	0.060	0.158	0.025	0.059	1.74	0.47
Experiment	0.168	0.90	2.13	0.080	0.88	1.78	0.148	0.034	0.067	0.114	0.036	0.076	2.10	0.68
Computed	0.105	1.83	3.93	0.139	1.72	3.85	0.196	0.082	0.213	0.140	0.099	0.223	2.90	0.82
Experiment	0.182	2.25	5.51	0.064	2.20	4.34	0.033	0.140	0.265	0.222	0.145	0.393	5.33	1.77
Computed	0.129	2.47	5.53	0.151	2.38	5.45	0.083	0.212	0.414	0.167	0.165	0.392	3.29	1.08
Experiment														
Computed	0.106	3.14	6.71	0.200	2.45	6.27	0.192	0.142	0.365	0.118	0.179	0.386	3.02	0.78
Experiment	0.101	3.92	8.18	0.500	3.86	7.47	0.140	0.199	0.447	0.084	0.197	0.402	5.50	1.68
Computed	0.126	1.60	3.66	0.097	1.75	3.63	0.146	0.090	0.209	0.089	0.106	0.261	2.05	0.69
Experiment	0.143	1.76	3.98	0.134	1.76	3.91	0.044	0.118	0.226	0.121	0.120	0.261	3.05	1.00
Computed	0.121	2.15	4.72	0.131	2.13	4.70	0.132	0.113	0.253	0.132	0.115	0.253	3.40	1.15
Experiment	0.177	2.39	5.78	0.094	2.36	4.87	0.062	0.132	0.259	0.272	0.135	0.424	7.20	2.40

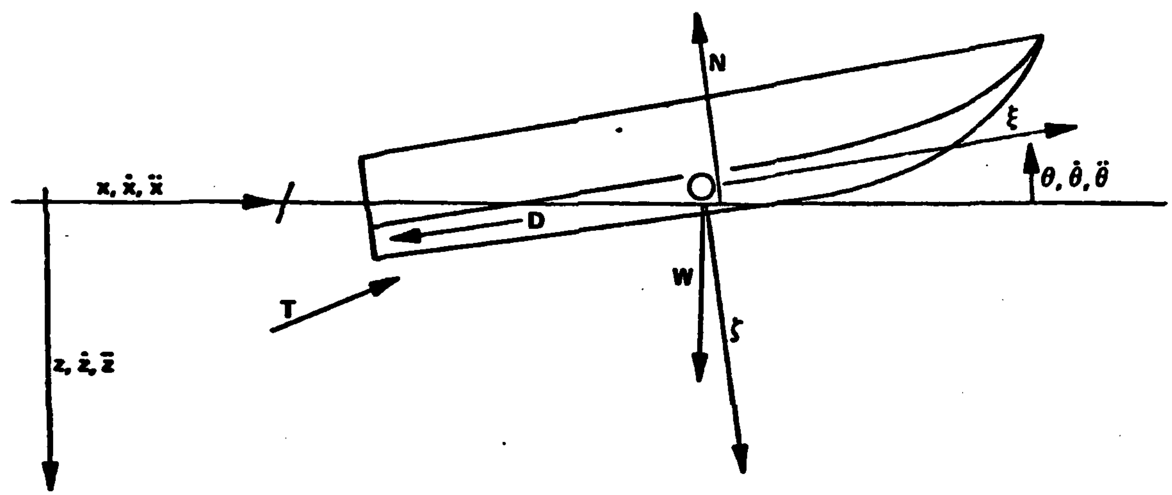


Figure 1 – Coordinate System

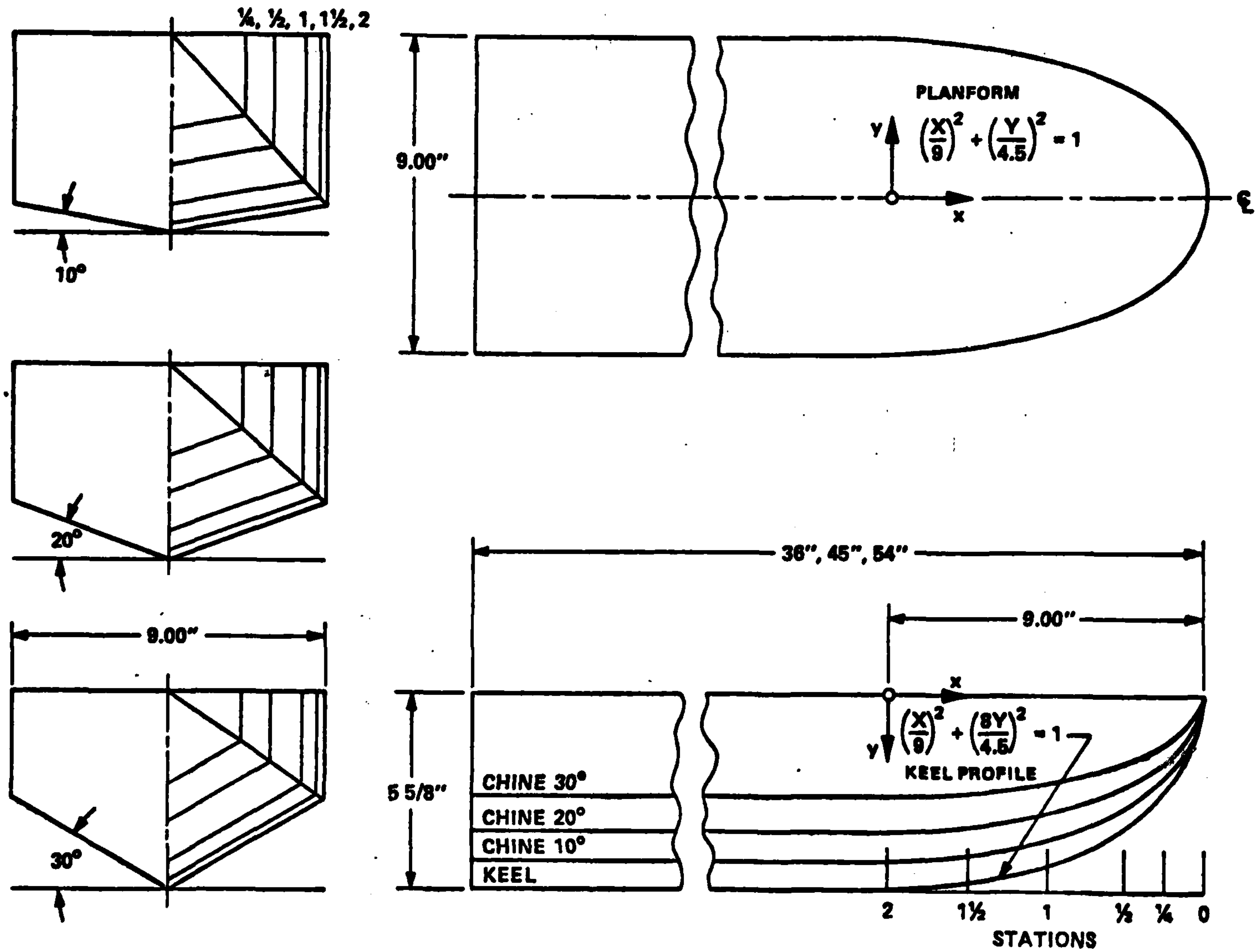


Figure 2 – Lines of Prismatic Models
(From Reference 3)

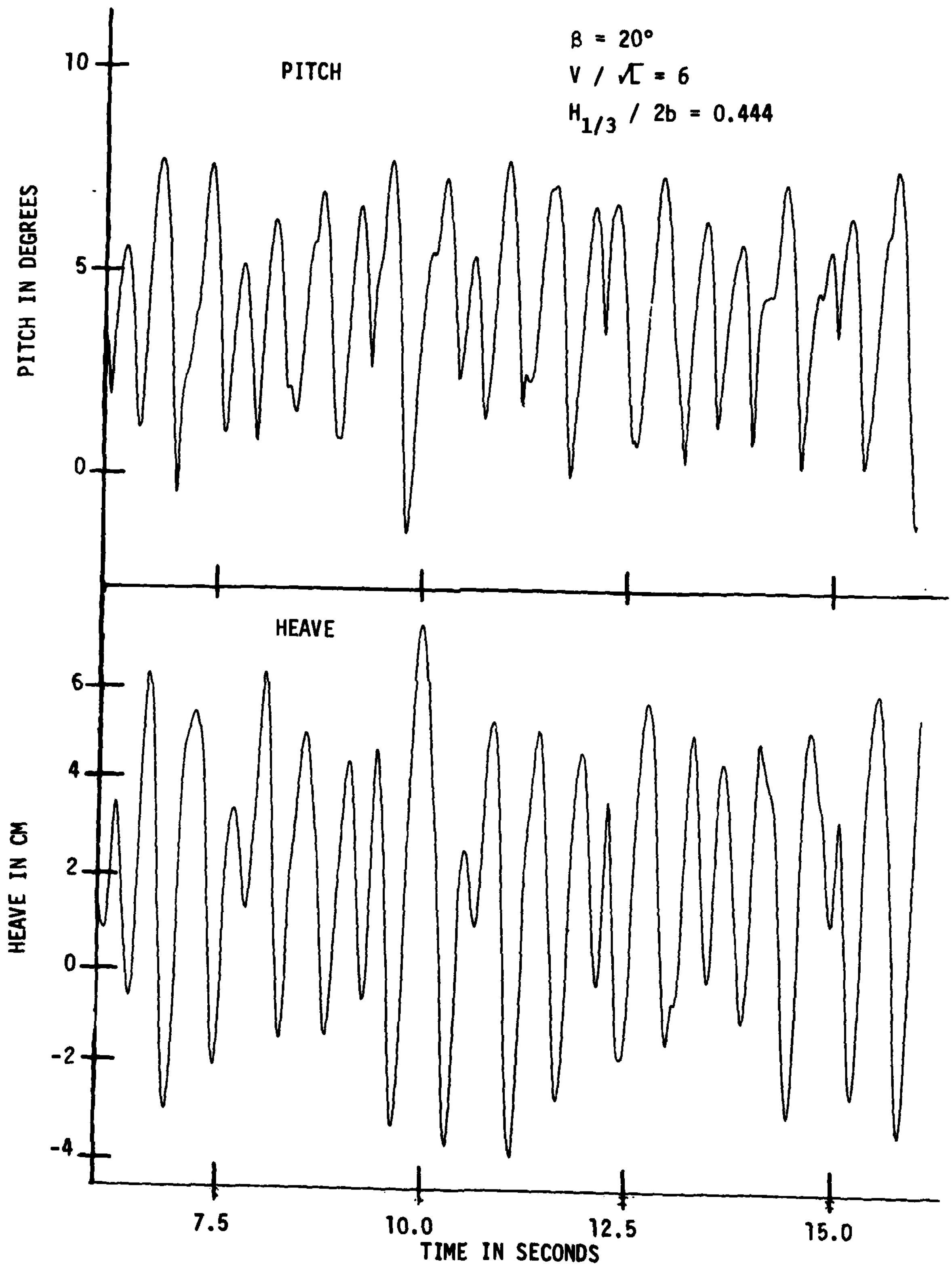
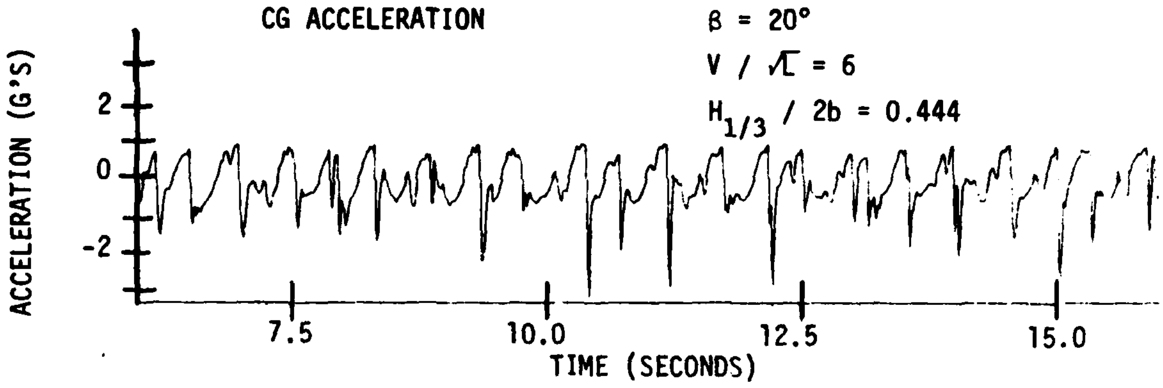


Figure 3 - Sample of Computed Pitch and Heave Motion



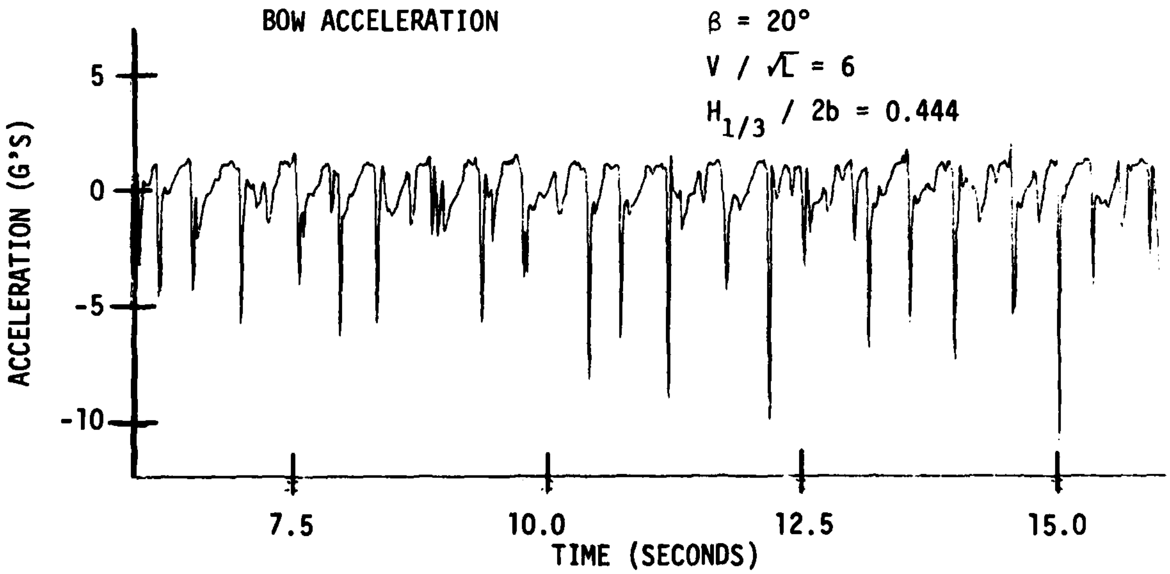


Figure 4 - Sample of Computed Bow and CG Acceleration

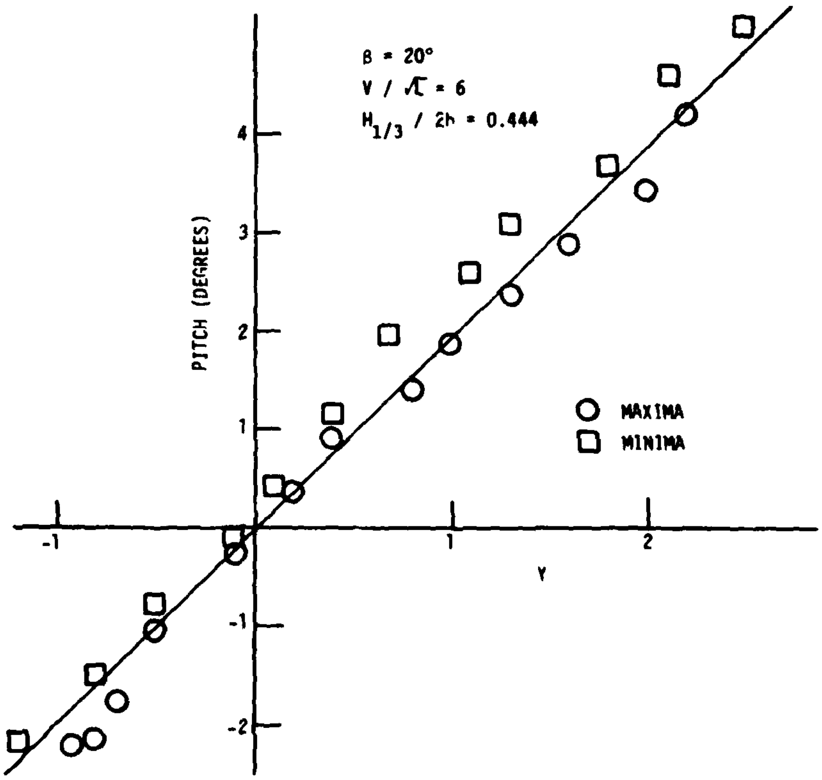


Figure 5 - Example of Pitch Motion Correlation with Generalized Rayleigh Distribution

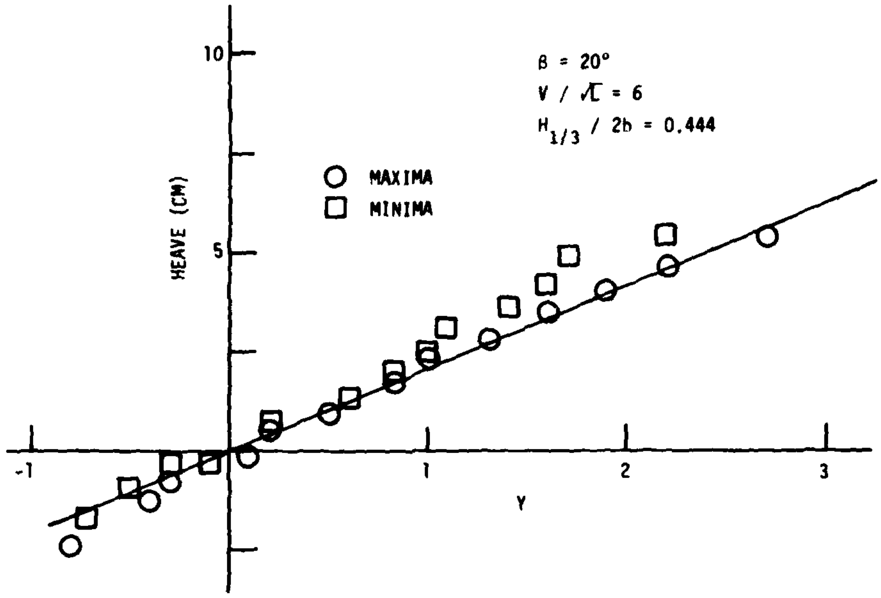


Figure 6 - Example of Heave Motion Correlation with Generalized Rayleigh Distribution

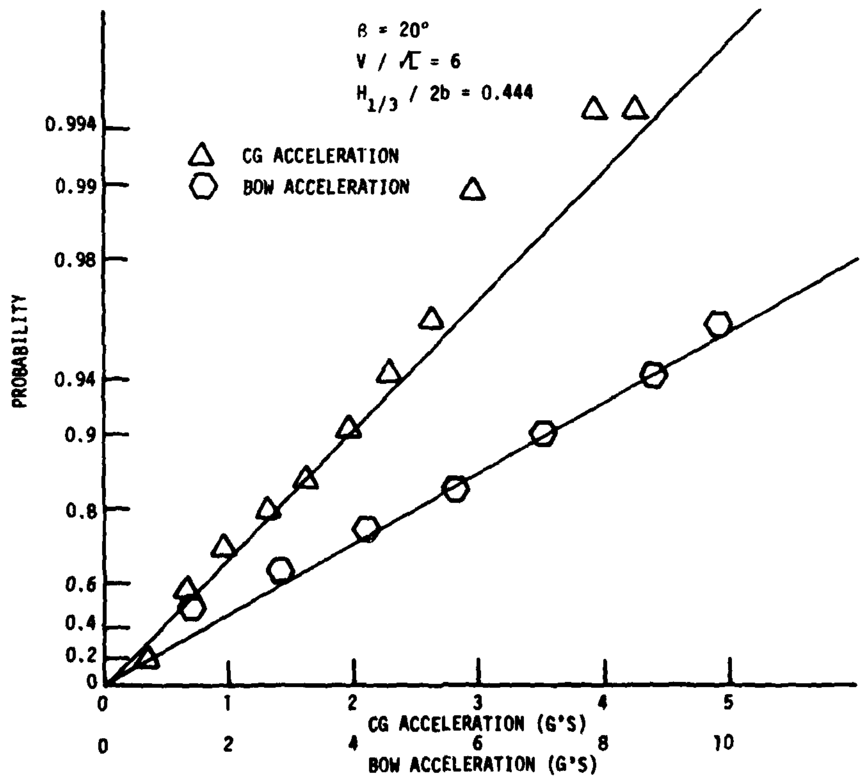


Figure 7 - Example of Bow and CG Acceleration Correlation with Exponential Distribution

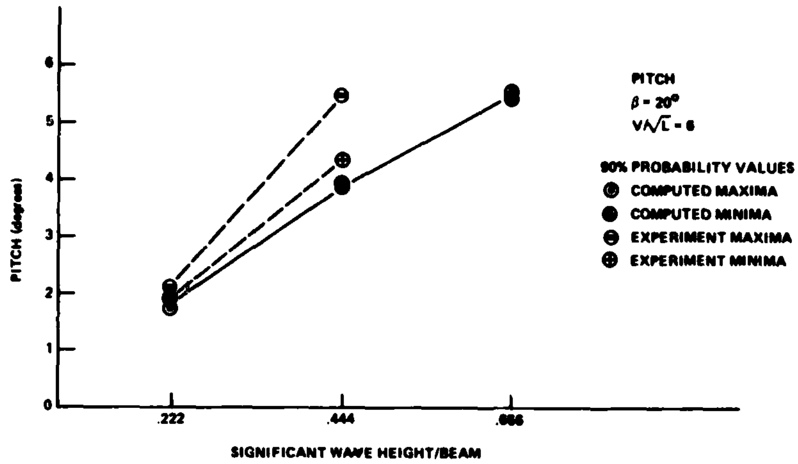


Figure 8 - Comparison of Computed and Experimental Pitch Variation with Significant Wave Height

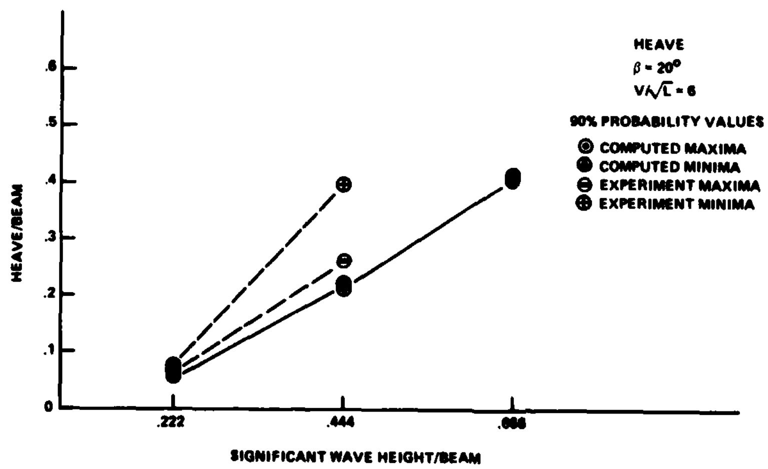


Figure 9 - Comparison of Computed and Experimental Heave Variation with Significant Wave Height

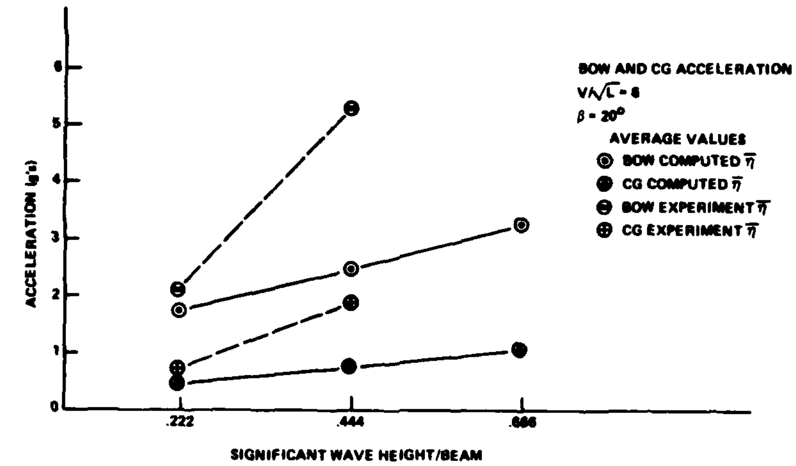


Figure 10 - Comparison of Computed and Experimental Bow and CG Acceleration Variation with Significant Wave Height

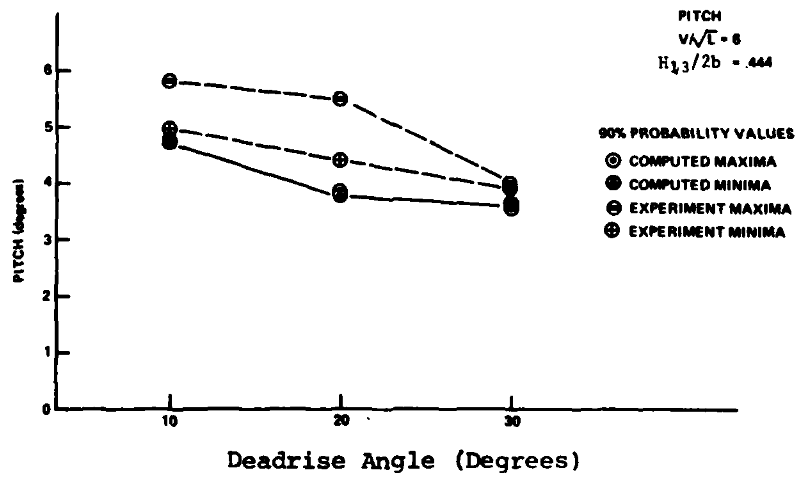


Figure 11 - Comparison of Computed and Experimental Pitch Variation with Deadrise Angle

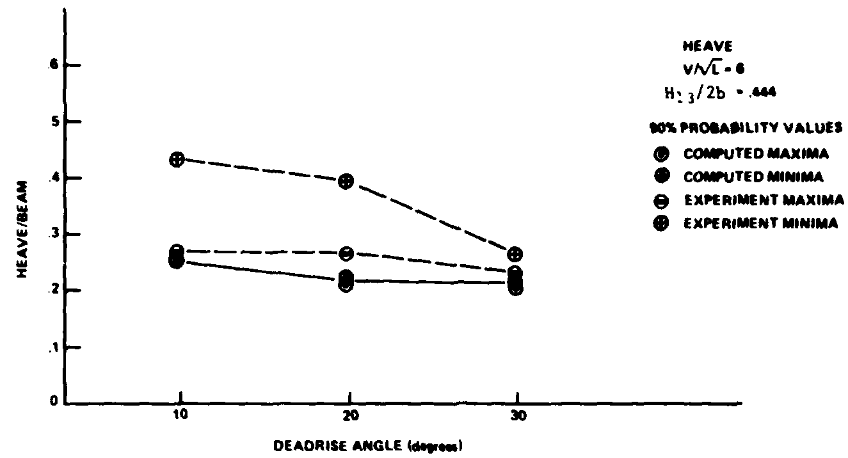


Figure 12 - Comparison of Computed and Experimental Heave Variation with Deadrise Angle

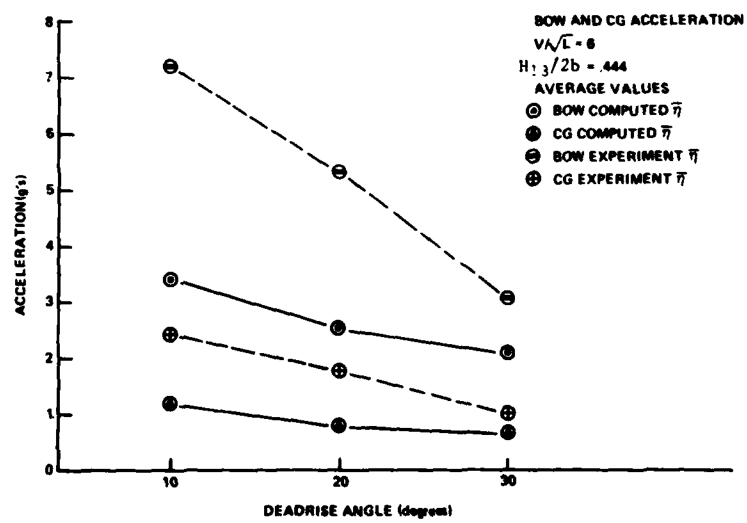


Figure 13 - Comparison of Computed and Experimental Bow and CG Acceleration Variation with Deadrise Angle

APPENDIX

LISTING OF COMPUTER PROGRAM FOR MOTION COMPUTATIONS

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3=512,
     .   TAPE2=512,TAPE4=512,TAPE9)
C
      REAL IT,M,MMAX,N,NO,NL,KAR
      INTEGER END
C
      DIMENSION X(6),FX(2,2000)
C
      COMMON /CONST/ NCG,ECG,PI,DPR,RPD,GRAVTY,RHO,NUM,MA(120),CD,TA,
     .         B(120),BETA,HW(120),TZ,DRAG,W,XD,T,XP,M,IT,
     1         DELTAS,TX,EST(120),KAR,MMAX(120),TEST(120),
     .              N(120),PHALF
      COMMON /SHIP/ MASS,CINT,QA,CE,CE2,CE3,DMU,EDMU,E2DMU,E3DMU,BF,BMM,
     .             NL,FL,IA,E(120)
      COMMON /IN/ BM(120),BI(120),VELIN
      COMMON/OUT/NPRINT,NPLOT,END
      COMMON/TERMS/T1,T2,T3,T4,T5,T6,T7,T8
      COMMON /TRANS/ START,RISE,RAMP
      COMMON /INTER/ II,KTT(10),DIFF(10)
      COMMON /IN2/ NO(120),XA,XE,HMAX,HMIN,A(6),EPSE(6)
      COMMON /ACCEL / XACCL,BWACL,CGACL,BL
C
      CALL INPUT
C
C        COMPUTE INTEGRATION INTERVAL INFORMATION
C
      NLESS = NUM-1
      I = 1
      II = 1
      DIFFER = EST(I+1)-EST(I)
      KTT(II) = 1
      DIFF(II) = DIFFER
      DO 25 I=2,NLESS
      DIFFER= EST(I+1)-EST(I)
      KTT(II) = KTT(II)+1
      IF(DIFFER.NE.DIFF(II))GO TO 24
      GO TO 25
   24 II = II+1
      KTT(II) = 1
      DIFF(II) = DIFFER
   25 CONTINUE
      KTT(II) = KTT(II)+1
C * * * * * CHECK IF NUMBER OF INTERVALS EXCEEDS DIMENSION
      IF (II.GT.10) WRITE(6,28) (KTT(I),DIFF(I),I=1,II)
      IF(II.GT.10) STOP 4
C * * * * * POINT AT WHICH MULTIPLE RUNS START
    8 CONTINUE
      CALL SEAWAY
      CALL TABLE
      TIME=XA
      KOUNT=1
      END=END-1
      WRITE(6,39)
   39 FORMAT(1H1)
C * * * * * * * * READ IN INITIAL CONDITIONS
C    X(1) = VELOCITY,  X(2) = Z DOT,  X(3) = THETA DOT
C     X(4) = X,         X(5) = Z,       X(6) = THETA
C      THETA IS READ IN DEGREES THEN CONVERTED TO RADIANS IN PROGRAM
C
```

```
      READ(5,10)(X(I),I=1,6)
C
C               DATA , USED IN RAMP FUNCTION, TO TURN ON WAVE
      READ(5,10)START,RISE
C
   10 FORMAT(8F10.4)
C * * * * * * * * * WRITE OUT THE INPUT VALUES
      WRITE(6,19) START,RISE,KAR
   19 FORMAT("     START = ",F10.4,/,"   RISE = ",F10.4,/,"   KAR = ",F10
     6.4)
C
C         TME IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS
C             TO BE CHANGED
C         HMX IS THE NEW MAXIMUM INTERVAL SIZE AFTER TIME TME
C         HMN IS THE NEW MINIMUM INTERVAL SIZE FOR KUTMER TO SUB-DIVIDE
C             THE MAXIMUM INTERVAL UP TO
C         IF THIS OPTION IS NOT USED SET TME TO THE STOP TIME OF THE RUN
C
      READ(5,10) TME,HMX,HMN
      WRITE(6,11) TME,HMAX,HMX,HMIN,HMN
   11 FORMAT(*  AT TIME *,F7.2,* THE MAXIMUM INTERVAL SIZE FOR INTEGRATI
     *ON WILL BE CHANGED FROM *,F10.4,* TO *,F10.4,/,
     **  AND THE MINIMUM SIZE FOR HALVING CHANGES FROM *,F10.4,
     *   * TO *,F10.4)
C         ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL
C         FOR CHECK AGAINST TIME IN THE INTEGARTION LOOP
            TM = TME-(HMAX/2.)
C
C
      IPT = 0
      IF(TME.EQ.XE) IPT = 1
C
      READ(5,10) PERCNT
      XACCL = ECG-PERCNT*BL
      WRITE(6,12) PERCNT,XACCL
   12 FORMAT(*  THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS
     *IS EQUAL TO ECG-*,F10.4,7H*BL OR ,F10.4)
C
      WRITE(6,23)
      WRITE(6,47)
   23 FORMAT(1H ,//)
   47 FORMAT(" STATION NO.",3X,"DEAD RISE",8X,"EST",8X,"NO",
     *   10X,"BEAM")
      WRITE(6,55)  ((I,BETA,EST(I),NO(I),BM(I)),I=1,NUM)
   55 FORMAT(6X,I2,5X,F10.4,4X,F10.4,4X,F10.4,3X,F10.4)
      WRITE(6,23)
      WRITE (6,56)  (X(I),I=1,6)
   56 FORMAT(" X VALUES",4X,6(F10.4,2X))
C * * * * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS
      X(3) = X(3)*RPD
      X(6) = X(6)*RPD
C
C * * * * * * * * * WRITE OUT COMPUTED ARRAYS
      WRITE(6,57)M,IT,PHALF,PI,GRAVIY
      IF(NPRINT.LT.4)  GO TO 62
      WRITE (6,58)  (E(I),I=1,NUM)
      WRITE (6,59)  (N(I),I=1,NUM)
      WRITE (6,64)  (PMAX(I),I=1,NUM)
      WRITE (6,65)  (TEST(I),I=1,NUM)
```

```
   62 CONTINUE
      WRITE(6,28)(KTT(I),DIFF(I),I=1,II)
   28 FORMAT(*  KTT,DIFF *,I10,2X,F10.4)
   57 FORMAT (4H M= ,F10.4,4H I= ,F10.4,11H PI*RHO/2= ,F10.4,
     1 5H PI= ,F10.4,10H GRAVITY= ,F10.4)
   58 FORMAT (" E(I)",10F10.4)
   59 FORMAT (" N(I)",10F10.4)
   64 FORMAT (" HMAX(I)",10F10.4)
   66 FORMAT (" TEST(I)",10F10.4)
      IB = 1
      IPRINT = NPRINT
      WRITE(4,91)
C * * * * * * * * WRITE HEADINGS AND CONDITIONS AT TIME = 0.
   91 FORMAT(1H1,2X,"TIME",9X,"XDOT",9X,"ZDOT",9X,"THETA DOT",6X,
     . 1HX,9X,1HZ,9X,5HTHETA,9X,2HNL,9X,2HFL,
     . 4X,8HBOW ACCL,4X,7HCG ACCL,//)
      WRITE(4,92) TIME,(X(I),I=1,6),NL,FL,BWACL,CGACL
      WRITE(9) TIME,(X(I),I=4,6),BWACL,CGACL
      KOUNT = KOUNT+1
      FX(1,IB)=X(5)
      FX(2,IB)=X(6)
      IKUTM = (TME-XA)/HMAX + (XE-TME)/HMX + .05
      FIRST=0.0
      NEQS=6
      IKUTS=0
C
C       START OF INTEGRATION LOOP
C
  851 CONTINUE
      NPRINT = IPRINT
C * * * * * * * * CHECK PITCH .GT. .5236 RADIANS
      IF(X(6).GT..5236)GO TO 853
C * * * * * * * *  PERFORM INTEGRATIONS
      IF(TIME.LT.TM.OR.TME.EQ.XE) GO TO 98
        IF(IPT.EQ.1) GO TO 98
          HMIN = HMN
          HMAX = HMX
          FIRST = 0.0
   98 CONTINUE
      CALL KUTMER(NEQS,TIME,HMAX,X,EPSE,A,HMIN,FIRST)
      IKUTS=IKUTS+1
      IF(FIRST.EQ.2)GO TO 861
      IF(KOUNT.NE.1.AND.KOUNT.NE.41) GO TO 99
      WRITE(4,91)
      KOUNT=1
C * * * * * * * *  WRITE OUT TIME INTERVAL RESULTS
   99 WRITE(4,92) TIME,(X(I),I=1,6),NL,FL,BWACL,CGACL
      WRITE(6,93)T1,T2,T3,T4,T5,T6,T7,T8,BMM,BF
      WRITE(9) TIME,(X(I),I=4,6),BWACL,CGACL
      IF(TIME.LT.TM.OR.TME.EQ.XE) GO TO 200
      IF(IPT.EQ.1) GO TO 200
      CALL PLOTER (FX,XA,HMAX,IB,IPT)
      IPT = 1
      IB = 0
      XA = TIME
      FIRST = 0.0
      HMIN = HMN
      HMAX = HMX
  200 CONTINUE
```

```
      IB=IB+1
      FX(1,IB)=X(5)
      FX(2,IB)=X(6)
   93 FORMAT(" ",10E10.4)
   92 FORMAT(1X,11(F10.4,2X))
  100 CONTINUE
      KOUNT=KOUNT+1
   21 CONTINUE
      IF(TIME.LE.XE.AND.IKUTS.LT.IKUTM)GO TO 851
      WRITE(2,852)
  854 CONTINUE
  852 FORMAT( "    END OF KUTMER")
  853 CONTINUE
      CALL PLOTER (FX,XA,HMAX,IB,IPT)
C * * * * * * *  CHECK FOR LAST RUN IF NOT CYCLE BACK TO READ
C     NEW DATA FOR NEXT RUN
      IF(END.NE.1)GO TO 8
      GO TO 999
C * * * * KUTMER ERROR MESSAGES
  861 WRITE(6,862)
  862 FORMAT("     ERROR CRITERION IN KUTMER CAN NOT BE MET")
      WRITE (6,56) (X(I),I=1,6)
      WRITE (6,86) TIME
   86 FORMAT (" TIME =",F10.4)
      IF(END.NE.1)GO TO 8
      GO TO 853
  999 CONTINUE
      END FILE 9
      END
      SUBROUTINE PLOT2(F,FMIN,FMAX,NVAR,NFUN,N1,N,X0,DELX)
C
C PLOT FIRST N POINTS OF UP TO 26 FUNCTIONS F(X)
C  F(I,J) CONTAINS THE VALUE FOR THE JTH POINT OF THE ITH FUNCTION
C  FMIN(I) AND FMAX(I) CONTAIN THE MIN AND MAX ORDINATE VALUES FOR
C          THE ITH FUNCTION.
C       NVAR(I)    AN ARRAY OF TITLES FOR THE VARIOUS FUNCTIONS
C                  TO BE PLOTTED AGAINST THE ABSCISSA
C       NFUN       NUMBER OF FUNCTIONS TO BE PLOTTED -  DIMENSION OF
C                  NVAR, FMIN, FMAX
C       N1         USED ONLY IN F(N1,1) AS PASSED DIMENSION
C       N          NUMBER OF POINTS IN A SINGLE PLOT FRAME
C       X0         FIRST ABSCISSA VALUE
C       DELX       ABSCISSA INCREMENT
C
      DIMENSION ASTEP(26),F(N1,N),FMIN(NFUN),FMAX(NFUN),VLAST(26),
     1        VFIRST(26),HEAD(6),STEP(26)
       INTEGER CH(26),NVAR(  NFUN),DOT,ASTER,PLUS,BLANK
       INTEGER C
       INTEGER A(101)
C
      DATA BLANK,DOT,ASTER,PLUS/1H ,1H.,1H*,1H+/
      DATA CH(1),CH(2),CH(3),CH(4),CH(5),CH(6),CH(7),CH(8),CH(9),CH(10)
     2   / 1HA , 1HB , 1HC , 1HD , 1HE , 1HF , 1HG , 1HH ,1HI ,1HJ /
      DATA CH(11),CH(12),CH(13),CH(14),CH(15),CH(16),CH(17),CH(18)
     2   /  1HK ,  1HL ,  1HM ,  1HN ,  1HO ,  1HP ,  1HQ ,  1HR/
      DATA CH(19),CH(20),CH(21),CH(22),CH(23),CH(24),CH(25),CH(26)
     2   /  1HS ,  1HT ,  1HU ,  1HV ,  1HW ,  1HX ,  1HY ,  1HZ /
C
      IF(NFUN.LE.0.OR.N.LE.0) RETURN
```

```
C PRINT HEADINGS.
      WRITE(6,46)
   46 FORMAT (///)
      DO 40 I=1,NFUN
 30    TENM=ABS(FMAX(I)-FMIN(I))
       EXP=1.
       IF (TENM.EQ.0.) GO TO 2
C BRING TENM TO A VALUE BETWEEN 1 AND 10
       IF(TENM.LT.1.) GO TO 1
 3     IF(TENM.LT.10.) GO TO 2
       EXP=EXP*10.
       TENM=TENM*.1
       GO TO 3
 1     EXP=EXP*.1
       TENM=TENM*10.
       IF(TENM.GT.1.) GO TO 2
       GO TO 1
C  SET UP VALUE BETWEEN GRID LINES. RSTEP.
 2     PSTEP=5.
       IF(TENM.GE.5.)PSTEP=10.
       IF(TENM.LT.2.)PSTEP=2.
 5     RSTEP(I)=PSTEP*EXP*.1
C  COMPUTE VALUE OF STARTING LINE. VFIRST.
       FIRST=FMIN(I)/RSTEP(I)
       IF(FMIN(I).LT.0.)FIRST=FIRST-1.
       FIRST=AINT(FIRST)
       VFIRST(I)=FIRST*RSTEP(I)
C  CHECK END LINE VALUE.VLAST.
       VLAST(I)=VFIRST(I)+10.*RSTEP(I)
       IF(VLAST(I).GT.FMAX(I))GO TO 4
C  IF GRAPH IS TOO SMALL TAKE NEXT LARGER STEP.
       AA=PSTEP
       IF(AA.LT.5.)PSTEP=5.
       IF(AA.EQ.5.)PSTEP=10.
       IF(AA.LT.10.) GO TO 5
       PSTEP=2.
       EXP=10.*EXP
       GO TO 5
C  COMPUTE VALUE BETWEEN POINTS.STEP.
 4     STEP(I)=RSTEP(I)*.1
       RK=0.
      DO 6 KK=1,6
       HEAD(KK)=VFIRST(I)+2.*RK*RSTEP(I)
 6     RK=RK+1.
  40   WRITE (6,45) CH(I), NVAR(I), (HEAD(KK),KK=1,6)
   45 FORMAT(1X,A1,3H = ,A10,5X,1PE12.4,5(8X,1PE12.4))
      DO 50 J=1,101
      A(J)=BLANK
      IF(MOD(J,10).EQ.1) A(J)=DOT
   50 CONTINUE
      WRITE(6,55) A,A
   55  FORMAT (25X,101A1/15X,4HTIME,6X,101A1)
C PLOT EACH POINT
      DO 100 J=1,N
      B=X0+FLOAT(J-1)*DELX
      DO 70 K=1,101
      A(K)=BLANK
      IF(MOD(K,10).EQ.1) A(K)=DOT
      IF(MOD(J,5 ).EQ.1) A(K)=DOT
```

```
   70 CONTINUE                                                      PLOT2 91
      DO 80 I=1,NFUN                                                PLOT2 92
       LOC=((F(I,J)-VFIRST(I))/STEP(I)+1.5)                         PLOT2 93
       C=A(LOC)                                                     PLOT2 94
       A(LOC)=CH(I)                                                 PLOT2 95
      IF(C.NE.BLANK.AND.C.NE.DOT) A(LOC)=ASTER                      PLOT2 96
   80 CONTINUE                                                      PLOT2 97
       IF(MOD(J,10).EQ.1)GO TO 95                                   PLOT2 98
      WRITE(6,85) A                                                 PLOT2 99
  85   FORMAT (25X,101A1)                                           PLOT2100
       GO TO 100                                                    PLOT2101
   95 WRITE(6,15)B,A                                                PLOT2102
   15  FORMAT (12X,1PE12.4,1X,101A1)                                PLOT2103
  100 CONTINUE                                                      PLOT2104
      RETURN                                                        PLOT2105
      END                                                           PLOT2106
      SUBROUTINE KUTMER(NO,T,H,Y0,EPSE,A,HCX,FIRST)                 KUTMER 2
      DIMENSION Y0(6),Y1(6),Y2(6),F0(6),F1(6),F2(5),EPSE(6),A(6)    KUTMER 3
      COMMON/OUT/NPRINT,NPLOT,END                                   KUTMER 4
      COMMON /ACCEL / XACCL,BWACL,CGACL,BL                          KUTMER 5
      DATA NAM1,NAM2 /2HY1,2HY2 /                                   KUTMER 6
C                                                                   KUTMER 7
C         NO = NUMBER OF EQUATIONS, NO. OF COMPONENTS OF Y0         KUTMER 8
C         T  = INDEPENDENT VARIABLE                                 KUTMER 9
C         H  = INCREMENT FOR WHICH SOLUTION IS TO BE RETURNED + OR - KUTMER10
C         Y0 = THE VECTOR OF DEPENDENT VARIABLES.  ENTER WITH INITIAL KUTMER11
C     VALUES AT T AND RETURN WITH VALUES AT T+H                      KUTMER12
C      EPSE = RELATIVE ERROR CRITERION FOR COMPONENTS OF Y0 .GT ABS(A) KUTMER13
C         A  = ABSOLUTE ERROR SRITERION FOR COMPONENTS OF Y0 .LT. ABS(A) KUTMER14
C  NOTE-- EPSE AND A MUST BE SPECIFIED FOR EACH COMPONENT OF THE SYSTEM KUTMER15
C        HCX = THE SMALLEST STEP SIZE USED IN THE INTEGRATION       KUTMER16
C    FIRST SHOULD BE 0 WHEN KUTMER IS ENTERED FOR THE FIRST TIME    KUTMER17
C    AFTER THAT FIRST IS 1 IF KUTMER IS ENTERED WITH THE SAME H OR  KUTMER18
C       IF IT IS ENTERED WITH A CHANGED H                           KUTMER19
C    IF FIRST IS 2 THE ERROR SRITERIA CANNOT BE MEET AND THE STEP SIZE IKUTMER20
C    REDUCED TO H/128.                                              KUTMER21
C                                                                   KUTMER22
      IF (FIRST) 20,10,20                                           KUTMER23
C - - - - - - - - - - - FIRST ENTRY                                 KUTMER24
   10 HC = H                                                        KUTMER25
      IPLOC = 1                                                     KUTMER26
      FIRST = 1.                                                    KUTMER27
C - - - - - - - - - -  OTHER ENTRY                                  KUTMER28
   20 LOC = 0                                                       KUTMER29
      HCX = HC                                                      KUTMER30
      IF (HC.NE.0.) GO TO 30                                        KUTMER31
      WRITE(6,800)                                                  KUTMER32
  800 FORMAT(5X,45HKUTMER ENTERED WITH ZERO INTEGRATION INTERVAL   ) KUTMER33
      FIRST = 2.                                                    KUTMER34
      RETURN                                                        KUTMER35
C - - - - - - - - - - -  5 CALLS TO DAUX                            KUTMER36
   30 CALL DAUX(T,Y0,F0)                                            KUTMER37
      IF(NPRINT.EQ.5)WRITE(6,400)Y0,T,F0                            KUTMER38
  400 FORMAT(6(2X,F10.4),4HTIME,2X,F10.4)                           KUTMER39
      IF(NPRINT.EQ.5)WRITE(6,400)HC                                 KUTMER40
   39 DO 40 I=1,NO                                                  KUTMER41
   40 Y1(I) = Y0(I)+(HC/3.)*F0(I)                                   KUTMER42
      IF(NPRINT.EQ.5)WRITE(6,400)Y1,T                               KUTMER43
C                                                                   KUTMER44
```

```
      CALL DAUX(T+HC/3.,Y1,F1)                                          KUTMER45
      IF(NPRINT.EQ.5)WRITE(6,400)F1,T                                   KUTMER46
      DO 50 I=1,ND                                                      KUTMER47
   50 Y1(I) = Y0(I)+(HC/6.)*F0(I)+(HC/6.)*F1(I)                         KUTMER48
      IF(NPRINT.EQ.5)WRITE(6,400)Y1,T                                   KUTMER49
C                                                                       KUTMER50
      CALL DAUX(T+HC/3.,Y1,F1)                                          KUTMER51
      IF(NPRINT.EQ.5)WRITE(6,400)F1,T                                   KUTMER52
      DO 60 I=1,ND                                                      KUTMER53
   60 Y1(I) = Y0(I)+(HC/8.)*F0(I)+.375*HC*F1(I)                         KUTMER54
      IF(NPRINT.EQ.5)WRITE(6,400)Y1,T                                   KUTMER55
C                                                                       KUTMER56
      CALL DAUX(T+HC/2.,Y1,F2)                                          KUTMER57
      IF(NPRINT.EQ.5)WRITE(6,400)F2,T                                   KUTMER58
      DO 70 I=1,ND                                                      KUTMER59
   70 Y1(I) = Y0(I)+(HC/2.)*F0(I)-1.5*HC*F1(I)+2.*HC*F2(I)              KUTMER60
      IF(NPRINT.EQ.5)WRITE(6,400)Y1,T                                   KUTMER61
C                                                                       KUTMER62
      CALL DAUX(T+HC,Y1,F1)                                             KUTMER63
      IF(NPRINT.EQ.5)WRITE(6,400)F1,T                                   KUTMER64
      DO 80 I=1,ND                                                      KUTMER65
   80 Y2(I) = Y0(I)+HC/6.*F0(I)+(2./3.)*HC*F2(I)+(HC/6.)*F1(I)          KUTMER66
      IF(NPRINT.EQ.5)WRITE(6,400)Y2,T                                   KUTMER67
      INC = 0                                                           KUTMER68
C - - - - - - - - - - -  CHECK ERROR CRITERIA                           KUTMER69
      DO 110 I=1,ND                                                     KUTMER70
      ZZZ = ABS(Y1(I))-A(I)                                             KUTMER71
      IF (ZZZ) 85,87,87                                                 KUTMER72
C - - - - - - - - - - -  ABSOLUTE ERROR                                 KUTMER73
   85 ERROR = ABS(.2*(Y1(I)-Y2(I)))                                     KUTMER74
      IF (ERROR-A(I)) 100,100,90                                        KUTMER75
C - - - - - - - - - - -   RELATIVE ERROR                                KUTMER76
   87 ERROR = ABS(.2-.2*Y2(I)/Y1(I))                                    KUTMER77
      IF(ERROR-EPSE(I)) 100,100,90                                      KUTMER78
C - - - - - - - - - - -  SINCE ERROR .GT. ERROR CRITERIA CHECK IF HC.GT.H/KUTMER79
C - - - - - - - - - - -  IF YES THEN HALVE INTERVAL.  OTHERWISE STOP.   KUTMER80
   90 X = 128.*ABS(HC)-ABS(H)                                           KUTMER81
      IF(X) 91,95,95                                                    KUTMER82
C - - - - - - - - - - -  ERROR TOO LARGE                                KUTMER83
   91 WRITE(6,92)I,T,ERROR,HC                                           KUTMER84
   92 FORMAT(/18H FOR EQUATION NO. I2,27H, THE RELATIVE ERROR AT T = ,  KUTMER85
     .    E15.8, 4H IS ,E15.8,13H STEP SIZE = ,E15.8)                   KUTMER86
      FIRST = 2.                                                        KUTMER87
      RETURN                                                            KUTMER88
C - - - - - - - - - - -  HALVE INTERVAL                                 KUTMER89
   95 HC = HC/2.                                                        KUTMER90
      IPLOC = 2*IPLOC                                                   KUTMER91
      LOC = 2*LOC                                                       KUTMER92
      HCX = HC                                                          KUTMER93
      WRITE(2,710)T,I,ERROR,HC                                          KUTMER94
  710 FORMAT(/8H TIME = ,F10.3,5X,26HHALVE INTERVAL.  EQUATION  ,I3,    KUTMER95
     ,13H HAS ERROR = ,E16.8,6X,17H STEP SIZE NOW = ,E15.8)             KUTMER96
      WRITE(2,720) NAM2,(Y2(J),J=1,ND)                                  KUTMER97
      WRITE(2,720) NAM1,(Y1(J),J=1,ND)                                  KUTMER98
  720 FORMAT( 2X,A2 / 3(10E13.5/))                                      KUTMER99
      GO TO 30                                                          KUTME100
C - - - - - - - - - - -  TEST IF INTERVAL LENGTH CAN BE DOUBLED         KUTME101
  100 IF (ERROR*64.-EPSE(I)) 110,110,101                                KUTME102
  101 INC = 1                                                           KUTME103
```

```
  110 CONTINUE
C - - - - - - - - - - -   UPDATE T AND SOLUTION
  111 T = T+HC
      DO 112 I=1,NO
  112 Y0(I) = Y2(I)
C - - - - - - - - - - -   GET SOLUTION IN NEXT INTERVAL
      LOC = LOC+1
      IF (LOC-IPLOC) 120,210,210
  120 IF(INC)210,130,210
  130 IF (LOC-(LOC/2)*2) 210,140,210
  140 IF(IPLOC-1)210,210,200
C - - - - - - - - - - -   DOUBLE INTERVAL LENGTH
  200 HC = 2.*HC
      LOC = LOC /2
      IPLOC = IPLOC/2
  210 IF(IPLOC-LOC) 30,329,30
  329 BWACL = F0(2)-XACCL*F0(3)
      CGACL = F0(2)
      RETURN
      END
      SUBROUTINE DAUX(TIME,X,RHS)
C
C          TIME       TIME AT WHICH SYSTEM IS TO BE EVALUATED
C          X          STATE VECTOR
C          RHS        THE RIGHT HAND SIDE OF THE EQUATION  S = F A
C
      REAL KAR
      REAL IA,IT,M,K,MA,MASS,NL,N,MMAX
      DIMENSION X(6),RHS(6),F(3,1),A(3,3),INDEX(3,3),
     1          Q(120),U(120),V(120),WOTIME(20)
C
      COMMON /SHIP/ MASS,CINT,QA,CE,CE2,CE3,DMU,EDMU,E2DMU,E3DMU,BF,BMM,
     .             NL,FL,IA,E(120)
      COMMON /CONST/ NCG,ECG,PI,DPR,RPD,GRAVTY,RHO,NUM,MA(120),CD,TA,
     .          B(120),BETA,HW(120),IZ,DRAG,W,XD,T,XP,M,IT,
     1          DELTAS,TX,EST(120),KAR,MMAX(120),TEST(120),
     .               N(120),PHALF
      COMMON /IN/ BM(120),B1(120),VELIN
      COMMON/OUT/NPRINT,NPLOT,END
      COMMON /TRANS/ START,RISE,RAMP
      COMMON /WAVE1/ ZMA,ZWMA,EMAS,ZZWMA,ZWEMA,Z2WMA,E2MAZ,
     1               ZWDOT(120),ASINPT(120,20),ACOSPT(120,20),CX6,SX6
      COMMON /WAVE2/ W0(20),K(20),C(20),RW0(20),RW02(20),RK(20),
     . R0(20),RWK(20),H,PHS(20)
C
      COMMON/SINE/POINT(1000)
      RAMP = RMP(TIME,START,RISE)
      PIH = PI/2.
      DO 5 JJ =1,10
      WOTIME(JJ) = W0(JJ)*TIME+PHS(JJ)
      PNTRAD = 318.3098861
    5 CONTINUE
      CX6 = COS(X(6))
      SX6 = SIN(X(6))
C*******SET VALUES OF MA AND B
      DO 75  I=1,NUM
      RI = 0.0
      PTO = X(4)+E(I)*CX6+N(I)*SX6
      VI = 0.0
```

```
      DO 20 J=1,10
      PT = PTO*K(J)+WOTIME(J)
      ANOX = PT*PNTRAD
      AS = AMOD(ANOX,2000.)-499.
      IF(AS.LT.-499.) AS = AS+2000.
      IS = AS
      IC = IS+500
      DX = AS-IS
      IF(IS.LE.0) IS = 2-IS
      IF(IS.GT.1000) IS = 2002-IS
      IF(IC.LE.0) IC = 2-IC
      IF(IC.GT.1000) IC = 2002-IC
      DX = DX/PNTRAD
      XSIN = POINT(IS)
      YCOS = POINT(IC)
      ASINPT(I,J) = XSIN+DX*YCOS
      ACOSPT(I,J) = YCOS-DX*XSIN
      RR = RO(J)*ACOSPT(I,J)
      VV = -RO(J)*K(J)*ASINPT(I,J)
      RI = RI+RR
      VI = VI+VV
   20 CONTINUE
      R(I) = RI*RAMP
C * * * * * * * *  COMPUTE HW SUBMERGENCE OF A POINT AND R  THE WAVE
C           HW(I) IS IN THE FIXED COORDINATE SYSTEM
      HW(I) = X(5)-E(I)*SX6+N(I)*CX6-R(I)
      IF(HW(I).GT.0)  GO TO 65
C              CRAFT IS NOT SUBMERGED
      MA(I) = 0.
      B1(I)=0.
      B(I) = 0.
      GO TO 75
   65 V(I) = VI*RAMP
      D(I) = HW(I)/(CX6-V(I)*SX6)
C             D(I) IS IN THE BODY AXIS SYSTEM AND IS THE SUBMERGENCE
      IF(D(I).GE.TEST(I))  GO TO 70
C              CRAFT IS PARTLY SUBMERGED
      B(I) = D(I)*(1./TA)*PIH
      B1(I) = D(I)*(1./TA)*PIH
      MA(I) = KAR*PHALF*B(I)*B(I)
      GO TO 75
C              CHINE IS IMMERSED
C                   B1 ARRAY IS USED FOR THE INTEGRALS OVER THE PORTION
C                   OF THE HULL FOR WHICH THE CHINE IS NOT IMMERSED
   70 MA(I)=MMAX(I)
      B(I)=BM(I)
      B1(I)=0.
   75 CONTINUE
      IF(NPRINT.LT.4)  GO TO 85
      WRITE(6,74)TIME
   74 FORMAT("  TIME = ",F10.4)
      WRITE(6,76)  (X(I),I=1,6)
      WRITE(6,77)  (R(I),I=1,NUM)
      WRITE (6,78) (HW(I),I=1,NUM)
      WRITE (6,79) ( B(I),I=1,NUM)
      WRITE (6,80) (V(I),I=1,NUM)
      WRITE (6,81) (D(I),I=1,NUM)
      WRITE(6,82)(MA(I),I=1,NUM)
   76 FORMAT("  X(I) ",6(2X,E12.6))
```

```
   77 FORMAT (" R(I)",10F10.4)
   78 FORMAT (" HW(I)",10F10.4)
   79 FORMAT (" H(I)",10F10.4)
   80 FORMAT (" V(I)",10F10.4)
   81 FORMAT (" D(I)",10F10.4)
   82 FORMAT(" MA(I) ",10F10.4)
   83 CONTINUE
C
C * * * * * * * * COMPUTES NL AND FL AND THE ASSOCIATED INTERGALS
      CALL FUNCT(X)
C
      IF(NPRINT.LT.4)GO TO 17
      WRITE(6,15) TX,FL,DRAG,TZ,W,NL,XD,T,XP
   15 FORMAT(" ",10E12.6)
   17 CONTINUE
C * * * * * * COMPUTE THE F VECTOR
      F(1,1) = TX+FL*SX6-DRAG*CX6
      F(1,1)=0.0
      F(2,1) = TZ+FL*CX6+DRAG*SX6+W
      F(3,1)=NL-DRAG*XD+T*XP
      IF(NPRINT.LT.3)GO TO 18
      WRITE(6,10)(F(I,1),I=1,3)
   18 CONTINUE
C * * * * * * COMPUTE THE A MATRIX
      A(1,1) = M+MASS*SX6*SX6
      A(1,2) = MASS*SX6*CX6
      A(1,3) = -DA*SX6
      A(1,2) = 0.
      A(1,3) = 0.
      A(2,1)=A(1,2)
      A(2,2) = M+MASS*CX6*CX6
      A(2,3) = -DA*CX6
      A(3,1)=A(1,3)
      A(3,2)=A(2,3)
      A(3,3)=IT+IA
      IF(NPRINT.LT.3)GO TO 25
      WRITE(6,12)(A(I,1),I=1,3)
      WRITE(6,13)(A(I,2),I=1,3)
      WRITE(6,14)(A(I,3),I=1,3)
C * * * * * * * INVERT THE A MATRIX
   25 CALL MATINS(A,3,3,F,1,1,DETERM,ID,INDEX)
      IF(ID.EQ.2)WRITE(6,26)
   26 FORMAT("     MATRIX IS SINGULAR          ")
C*******A ON RETURN WILL CONTAIN THE INVERSE MATRIX
C         ID=2 MATRIX IS SINGULAR
C           =1 INVERSE WAS FOUND
C
C * * * * * * *  COMPUTE THE RIGHT HAND SIDE
      RHS(1) = F(1,1)
      RHS(2) = F(2,1)
      RHS(3) = F(3,1)
      RHS(1) = 0.0
      RHS(4) = X(1)
      RHS(5) = X(2)
      RHS(6) = X(3)
   10 FORMAT("    F(I,1) ",3(2X,E12.4))
   12 FORMAT("    A(I,1) ",3(2X,E12.4))
   13 FORMAT("    A(I,2) ",3(2X,E12.4))
   14 FORMAT("    A(I,3) ",3(2X,E12.4))
```

```
30 IF(NPRINT.LT.2) GO TO 40
   WRITE(6,12) (A(I,1),I=1,3)
   WRITE(6,13) (A(I,2),I=1,3)
   WRITE(6,14) (A(I,3),I=1,3)
   WRITE(6,35) (RHS(I),I=1,6)
35 FORMAT("   RHS(I) ",6(2X,E12.6))
40 CONTINUE
   RETURN
   END
   SUBROUTINE FUNCT(X)
   REAL KAR
   REAL IA,IAA,IPART,KPI,MA,MASS,N
   INTEGER END
   DIMENSION IPART(120),C1(120),C2(120),
  &          D1(120),D2(120),D3(120),D4(120),D5(120),D6(120),
  .          QPART(120),Z1(120),Z2(120),Z3(120),Z4(120),Z5(120),
  .             Z6(120),Z7(120)
  .          ,X(6),VMAA(120)
C
   COMMON /SHIP/ MASS,CINT,QA,CE,CE2,CE3,DMU,EDMU,E2DMU,E3DMU,BF,BMM,
  &             NL,FL,IA,E(120)
   COMMON /CONST/ NCG,ECG,PI,DPR,RPD,GRAVTY,RHO,NUM,MA(120),CD,TA,
  .        B(120),BETA,HW(120),TZ,DRAG,W,XD,T,XP,M,IT,
  1        DELTAS,TX,EST(120),KAR,MMAX(120),TEST(120),
  .             N(120),PHALF
   COMMON /IN/ HM(120),BI(120),VELIN
   COMMON/OUT/NPRINT,NPLOT,END
   COMMON /WAVE1/ ZMA,ZWMA,EMAS,ZZWMA,ZWEMA,Z2WMA,E2MAZ,
  1               ZWDOT(120),ASINPT(120,20),ACOSPT(120,20),CX6,SX6
   COMMON /WAVE2/ W0(20),K(20),C(20),RW0(20),RW02(20),RK(20),
  . RO(20),RWK(20),H,PHS(20)
   COMMON /INTER/ II,KTT(10),DIFF(10)
   COMMON/TRANS/START,RISE,RAMP
   COMMON /TEST/ VMA
C * * * * * * * * * * INITIALIZE INTEGRAL SUMS
   MASS = 0.0
   QA = 0.0
   IA = 0.0
   CE = 0.0
   CE2 = 0.0
   DMU = 0.0
   EDMU=0.0
   E2DMU = 0.0
   E3DMU = 0.0
   BF = 0.0
   BMM = 0.0
   ZMA = 0.0
   ZWMA = 0.0
   EMAS = 0.0
   ZZWMA = 0.0
   ZWEMA = 0.0
   Z2WMA = 0.0
   E2MAZ = 0.0
   VPART = X(1)*SIN(X(6))+X(2)*COS(X(6))
C * * * * SET UP FUNCTIONS FOR INTEGRALS * * *  * * *
   DO 90 I=1,NUM
   IPART(I)=E(I)*E(I)*MA(I)
   QPART(I)=E(I)*MA(I)
   ZWDT = 0.0
```

```
      DWDT = 0.0
      DRDT = 0.0
      DWDE = 0.0
      XNE = X(3)*(N(I)*CX6-E(I)*SX6)
      XNEO = X(3)*(E(I)*CX6+N(I)*SX6)
      DO 15 J = 1,10
      ZWDTT = -RWO(J)*ASINPT(I,J)
      ZWDT = ZWDT+ZWDTT
      DWDTT = -RWO2(J)*ACOSPT(I,J)
      DWDT = DWDT+DWDTT
      DRDTT = -RK(J)*ASINPT(I,J)*(X(1)+C(J)+XNE)
      DRDT = DRDT+DRDTT
      DWDEE = -RWK(J)*ACOSPT(I,J)
      DWDE = DWDE+DWDEE
   15 CONTINUE
      ZWDOT(I) = ZWDT*RAMP
      DWDE = DWDE*RAMP
      DRDT = DRDT*RAMP
      U = X(1)*CX6-X(2)*SX6+ZWDOT(I)*SX6
      VEL = VPART-X(3)*E(I)-ZWDOT(I)*CX6
      Z1(I) = MA(I)*ZWDOT(I)
      Z2(I) = MA(I)*DWDT*RAMP
      Z3(I) = E(I)*Z2(I)
      Z4(I) = E(I)*Z1(I)
      Z5(I) = MA(I)*U*DWDE
      Z6(I) = E(I)*Z5(I)
      Z7(I) = MA(I)*VEL*U
      IF (VEL.LE.0.) GO TO 60
      IF (B1(I).LE.0.0) GO TO 50
      D1(I) = VEL*B1(I)*(X(2)-XNEO-DRDT)
      GO TO 51
   50 D1(I) = 0.
   51 CONTINUE
      D2(I) = E(I)*D1(I)
      C1(I) = VEL*VEL*H(I)
      C2(I) = E(I)*C1(I)
      GO TO 61
   60 D1(I) = 0.
      D2(I) = 0.
      C1(I) = 0.
      C2(I) = 0.
   61 CONTINUE
      D3(I) = MA(I)*DWDE*VEL
      D4(I) = E(I)*D3(I)
      PIH = PI/2.
      D5(I) = B(I)*(HW(I)-B(I)*TA/2.)
   65 D6(I) = D5(I)*E(I)*.5
   90 CONTINUE
      RHOG=RHO*GRAVTY
      PIH = PI/2.
      KPI = KAR*PI
C     EVALUATE INTEGRALS USING TRAP METHOD
      I = 1
      INDEX = 1
   91 CALL TRAP(MA(INDEX),DIFF(I),KTT(I),TMASS)
      CALL TRAP(VPART(INDEX),DIFF(I),KTT(I),OA1)
      CALL TRAP(C1(INDEX),DIFF(I),KTT(I),CEA)
      CALL TRAP(C2(INDEX),DIFF(I),KTT(I),CE2A)
      CALL TRAP(IPART(INDEX),DIFF(I),KTT(I),IAA)
```

```
      CALL TRAP(D1(INDEX),DIFF(I),KTT(I),DMUA)
      CALL TRAP(D2(INDEX),DIFF(I),KTT(I),EDMUA)
      CALL TRAP(D3(INDEX),DIFF(I),KTT(I),E2DMUA)
      CALL TRAP(D4(INDEX),DIFF(I),KTT(I),E3DMUA)
      CALL TRAP(D5(INDEX),DIFF(I),KTT(I),BFA)
      CALL TRAP(D6(INDEX),DIFF(I),KTT(I),BMMA)
      CALL TRAP(Z1(INDEX),DIFF(I),KTT(I),ZMAA)
      CALL TRAP(Z2(INDEX),DIFF(I),KTT(I),ZWMAA)
      CALL TRAP(Z3(INDEX),DIFF(I),KTT(I),EMASA)
      CALL TRAP(Z4(INDEX),DIFF(I),KTT(I),ZZWMAA)
      CALL TRAP(Z5(INDEX),DIFF(I),KTT(I),ZWEMAA)
      CALL TRAP(Z6(INDEX),DIFF(I),KTT(I),Z2WMAA)
      CALL TRAP(Z7(INDEX),DIFF(I),KTT(I),E2MAZA)
C
   93 CONTINUE
      MASS = MASS + TMASS
      QA = QA + QA1
      IA = IA + IAA
      CE = CE + CEA
      CE2 = CE2 + CE2A
      DMU = DMU + DMUA
      EDMU = EDMU + EDMUA
      E2DMU = E2DMU + E2DMUA
      E3DMU = E3DMU + E3DMUA
      BF = BF + RHOG*BFA
      BMM = BMM + RHOG*BMMA
      ZMA = ZMA+ZMAA
      ZWMA = ZWMA+ZWMAA
      EMAS = EMAS+EMASA
      ZZWMA = ZZWMA+ZZWMAA
      ZWEMA = ZWEMA+ZWEMAA
      Z2WMA = Z2WMA+Z2WMAA
      E2MAZ = E2MAZ+E2MAZA
   94 CONTINUE
      IF ( I.GE.II)GO TO 92
      INDEX = INDEX+KTT(I)-1
      I = I+1
      GO TO 91
   92 CONTINUE
C
C * * * * * * * CALL COMPUT TO FIND THE VALUE OF NL AND FL USING
C     THE VALUES OF THE ABOVE INTEGRALS
      CALL COMPUT(X)
C
      IF(NPRINT.LT.3) GO TO 111
      IF(NPRINT.EQ.3) GO TO 108
      IF(NPRINT.EQ.4)GO TO 108
      WRITE(6,97)  (IPART(I),I=1,NUM)
      WRITE(6,98)  (QPART(I),I=1,NUM)
      WRITE(6,99)  (C1(I),I=1,NUM)
      WRITE(6,100)  (C2(I),I=1,NUM)
      WRITE(6,102)  (D1(I),I=1,NUM)
      WRITE(6,103)  (D2(I),I=1,NUM)
      WRITE(6,104)  (D3(I),I=1,NUM)
      WRITE(6,105)  (D4(I),I=1,NUM)
      WRITE(6,106)  (D5(I),I=1,NUM)
      WRITE(6,112)  (D6(I),I=1,NUM)
      WRITE(6,113)(Z1(I),I=1,NUM)
      WRITE(6,114)(Z2(I),I=1,NUM)
```

```
      WRITE(6,115)(Z3(I),I=1,NUM)                                      FUNCT170
      WRITE(6,116)(Z4(I),I=1,NUM)                                      FUNCT171
      WRITE(6,118)(Z5(I),I=1,NUM)                                      FUNCT172
      WRITE(6,119)(Z6(I),I=1,NUM)                                      FUNCT173
      WRITE(6,120)(Z7(I),I=1,NUM)                                      FUNCT174
      WRITE(6,107)KPI,RHOG,PIH                                         FUNCT175
  108 WRITE(6,109) MASS,CINT,QA,CE,CE2,CE3                             FUNCT176
      WRITE(6,121)IA                                                   FUNCT177
  121 FORMAT("  IA   ",E10.4)                                          FUNCT178
      WRITE(6,110)DMU,EDMU,E2DMU,E3DMU,BF,BMM                          FUNCT179
      WRITE(6,117)ZMA,ZWMA,EMAS,ZZWMA,ZWEMA,ZZWMA,E2MAZ                FUNCT180
   96 FORMAT("  CPART(I)",10(2X,E10.4))                                FUNCT181
   97 FORMAT("  IPART(I)",10(2X,E10.4))                                FUNCT182
   98 FORMAT("  QPART(I)",10(2X,E10.4))                                FUNCT183
   99 FORMAT("  C1      ",10(2X,E10.4))                                FUNCT184
  100 FORMAT("  C2      ",10(2X,E10.4))                                FUNCT185
  101 FORMAT("  C3      ",10(2X,E10.4))                                FUNCT186
  102 FORMAT("  D1      ",10(2X,E10.4))                                FUNCT187
  103 FORMAT("  D2      ",10(2X,E10.4))                                FUNCT188
  104 FORMAT("  D3      ",10(2X,E10.4))                                FUNCT189
  105 FORMAT("  D4      ",10(2X,E10.4))                                FUNCT190
  106 FORMAT("  D5      ",10(2X,E10.4))                                FUNCT191
  112 FORMAT("  D6      ",10(2X,E10.4))                                FUNCT192
  107 FORMAT(" KPHI ",E10.4," RHOG ",E10.4," PHIM ",E10.4)             FUNCT193
  109 FORMAT(" MASS ",E10.4," CINT ",E10.4," QA ",E10.4," CE ",E10.4,  FUNCT194
     *" CE2 ",E10.4," CE3 ",E10.4)                                     FUNCT195
  110 FORMAT(" DMU ",E10.4," EDMU ",E10.4," E2DMU ",E10.4," E3DMU ",   FUNCT196
     *E10.4," BF ",E10.4," BMM ",E10.4)                                FUNCT197
  113 FORMAT(4H Z1 ,10(2X,E10.4))                                      FUNCT198
  114 FORMAT(4H Z2 ,10(2X,E10.4))                                      FUNCT199
  115 FORMAT(4H Z3 ,10(2X,E10.4))                                      FUNCT200
  116 FORMAT(4H Z4 ,10(2X,E10.4))                                      FUNCT201
  118 FORMAT(4H Z5 ,10(2X,E10.4))                                      FUNCT202
  119 FORMAT(4H Z6 ,10(2X,E10.4))                                      FUNCT203
  120 FORMAT(4H Z7 ,10(2X,E10.4))                                      FUNCT204
  117 FORMAT(5H ZMA ,E10.4,6H ZWMA ,E10.4,6H EMAS ,E10.4,              FUNCT205
     .       7H ZZWMA ,E10.4,7H ZWEMA ,E10.4,7H ZZWMA ,E10.4,          FUNCT206
     .          7H E2MAZ ,E10.4)                                       FUNCT207
  111 CONTINUE                                                         FUNCT208
      RETURN                                                           FUNCT209
      END                                                              FUNCT210
      SUBROUTINE COMPUT(X)                                             COMPUT 2
      DIMENSION X(6)                                                   COMPUT 3
      REAL KAR,KPI                                                     COMPUT 4
      REAL NL,MASS,MA                                                  COMPUT 5
C                                                                      COMPUT 6
      COMMON /SHIP/ MASS,CINT,QA,CE,CE2,CE3,DMU,EDMU,E2DMU,E3DMU,BF,BMM,COMPUT 7
     .             NL,FL,IA,E(120)                                     COMPUT 8
      COMMON /CONST/ NCG,ECG,PI,DPR,RPD,GRAVTY,RHO,NUM,MA(120),CD,TA,  COMPUT 9
     .          H(120),BETA,HW(120),TZ,DRAG,W,XD,T,XP,M,IT,            COMPUT10
     1          DELTAS,TX,EST(120),KAR,MMAX(120),TEST(120),            COMPUT11
     .             N(120),PHALF                                        COMPUT12
      COMMON/OUT/NPRINT,NPLOT,END                                      COMPUT13
      COMMON /TERMS/ T1,T2,T3,T4,T5,T6,T7,T8                           COMPUT14
      COMMON /WAVE1/ ZMA,ZWMA,EMAS,ZZWMA,ZWEMA,ZZWMA,E2MAZ,            COMPUT15
     1             ZWOUT(120),ASINPT(120,20),ACOSPT(120,20),CX6,SX6    COMPUT16
      COMMON /TEST/ VMA                                                COMPUT17
C                                                                      COMPUT18
C * * * * * * * * SEE PAGES 7,8, AND 9 OF NOTES                        COMPUT19
```

```
C
      CX6 = COS(X(6))
      SX6 = SIN(X(6))
      PIH = PI/2.0
      KPI = KAR*PI
      CONS1 = CX6
      CONS2 = (KPI*RHO*PIH/TA)/CX6
      CONS3 = CX6*SX6
      CONS4 = CX6*CX6
      TERM1 = X(1)*CX6
      TERM2 = X(2)*SX6
      UVNUM = (X(1)*CX6-(X(2)-ZWDOT(NUM))*SX6)*
     &        (X(1)*SX6-X(3)*E(NUM)+(X(2)-ZWDOT(NUM))*CX6)
C
      ZMA = ZMA*X(3)*SX6
      ZZWMA = ZZWMA*X(3)*SX6
      ZWMA = ZWMA*CONS1
      EMAS = EMAS*CONS1
      DMU = DMU*CONS2
      EDMU = EDMU*CONS2
      CE = CE*CD*RHO
      CE2 = CE2*CD*RHO
      E2DMU = E2DMU*CONS3
      E3DMU = E3DMU*CONS3
      ZWEMA = ZWEMA*CONS4
      Z2WMA = Z2WMA*CONS4
C
   20 T1 = QA*X(3)*(TERM1-TERM2)
      T1 = T1 + ZZWMA - EMAS
      T2 = EDMU
      T3 = CE2
      T4 = MA(NUM)*E(NUM)*UVNUM + E2MAZ + E3DMU - Z2WMA + BMM
      NL = T1 + T2 + T3 + T4 + BMM
      T5 = MASS*X(3)*(TERM2-TERM1)
      T5 = T5 + ZWMA - ZMA
      T6 = -DMU
      T7 = -CE
      T8 = -MA(NUM)*UVNUM - E2DMU + ZWEMA
      BF = BF/CX6
C
      FL=T5+T6+T7+T8-BF
C
      IF(NPRINT.LT.3)GO TO 30
   25  CONTINUE
      WRITE(6,10)NL,FL
   10 FORMAT("  NL = ",E12.6,"  FL = ",E12.6)
   30 RETURN
      END
      SUBROUTINE INPUT
C* * * * * * * DEFINITION OF INPUT VARIABLES
C           XA = INITIAL TIME
C           XE = FINAL TIME
C         HMIN = MINIMUM STEP SIZE
C         HMAX = MAXIMUM STEP SIZE
C         EPSE = RELATIVE ERROR CRITERION USED FOR VALUES OF Y GT A
C        EPS = ERROR CRITERION IN KUTMER
C          A = ABSOLUTE ERROR CRITERIA USED IN KUTMER
C          NPRINT = 1  FINAL PRINTOUT
C                 = 2 MATRIX INVERSE MATRIX,F COLUMN MATRIX,AND KUTMER
```

```
C                    RESULTS
C                = 3 INTEGRAL VALUES
C                = 4 CALCULATED VALUES-CONSTANT FOR GIVEN INPUT VALUES
C       NPLOT = 0    NO PLOT
C             = 1  PRINTER PLOT
C         END = NUMBER OF RUNS
C
C           M = MASS OF CRAFT
C           W = WEIGHT OF CRAFT
C          TZ = THRUST COMPONENT IN Z DIRECTION
C          TX = THRUST COMPONENT IN X DIRECTION
C        XECG = DISTANCE FROM CG TO CENTER OF PRESSURE FOR NORMAL FORCE
C          XP = MOMENT ARM OF PROPELLER THRUST
C          XD = DISTANCE FROM CG  TO CENTER OF PRESSURE FOR DRAG FORCE
C       KA(I)= ADDED MASS COEFFICIENT
C         AN ARRAY GIVEN THE VALUE KAR WHICH IS READ IN
C       BM(I) = BEAM AT FREE SURFACE OR AT CHINE
C        DRAG = FRICTION DRAG
C           K = WAVE NUMBER
C          RO = WAVE HEIGHT
C          NU = WAVE SLOPE
C         NUM = NUMBER OF STATIONS
C          BL = BOAT LENGTH
C      LAMBDA = WAVE LENGTH
C          RG = RADIUS OF GENERATION IN FEET
C           T = PROPELLED THRUST IN LBS
C       GAMMA = PROPELLER THRUST ANGLE IN DEGREES
C       DELTAS=STATION SPACING IN FEET
C         ECG = LONGITUDINAL CENTER OF GRAVITY
C         NCG = VERTICAL CG
C     BETA(I) = DEAD RISE
C       NO(I) = HEIGHT OF MEAN BUTTOCK
C         RHO = DENSITY OF WATER
C      GRAVTY = GRAVITY FT/SEC**2
C         DPR = DEGREES PER RADIAN
C         RPD = RADIANS PER DEGREE
C          PI = 3.14159 . . . . . .
C     EST(I) = STATION POSITION
C     START = START TIME OF THE RAMP FUNCTION FOR SEA WAVE
C       RISE = DURATION OF THE RISE FROM ZERO TO ONE OF THE RAMP
C
C * * * * * * * * IC OPTIONS
C
C          IC(1)   =1 USE WAVE Z DISTANCE IN COMPUTING LIFT COMPONENT
C                     OF NL AND FL
C
C
      REAL IT,K,M,MA,MMAX,NU,N,NCG,NO,MASS,NL,IA,KAR
      INTEGER END
C
C
      COMMON /CONST/ NCG,ECG,PI,DPR,RPD,GRAVTY,RHO,NUM,MA(120),CD,TA,
     *          B(120),BETA,HW(120),TZ,DRAG,W,XD,T,XP,M,IT,
     1          DELTAS,TX,EST(120),KAR,MMAX(120),TEST(120),
     *                N(120),PHALF
      COMMON /SHIP/ MASS,CINT,OA,CE,CE2,CE3,OMU,EOMU,E2OMU,E3OMU,BF,BMM,
     *               NL,FL,IA,E(120)
      COMMON /IN/ HM(120),B1(120),VELIN
      COMMON /IN2/ NO(120),XA,XE,HMAX,HMIN,A(6),EPSE(6)
```

```
      COMMON/OUT/NPRINT,NPLOT,END
      COMMON /ACCEL/ XACCL,BWACL,CGACL,BL
C
      NAMELIST/HSP/A,NPRINT,NPLOT,END,W,BL,TZ,TX,XECG,XP,XD,
     1               DRAG,RG,T,GAMMA,ECG,NCG,KAR,NUM,BETA,EST
     *               ,XA,XE,HMIN,HMAX,EPS,VELIN
C
      DATA A /.01,.0001,.00001,.1,.0001,.00001/
      DATA NPRINT,NPLOT,END/1,1,1/
      DATA W,BL,TZ,TX,XECG,XP,XD,DRAG,RG,T,GAMMA,
     1    ECG,NCG,KAR /16.,3.75,6*0.0,.9526,2*0.0,
     *               2.325,0.0,1.0/
      DATA NUM,BETA,EST /77,20.0,
     6   0.0000,.03125,.06250,.09375,.12500,.15625,.18750,.21875,
     *   .25000,.28125,.31250,.34375,.37500,.40625,.43750,.46875,
     *   .50000,.53125,.56250,.59375,.62500,.65625,.6875,.71875,
     *   .75000,.78125,.81250,.84375,.87500,.90625,.93750,.96875,1.000,
     *   1.06250,1.12500,1.18750,1.25000,1.3125,1.37500,1.4375,
     *   1.500,1.5625,1.625,1.6875,1.75,1.8125,1.875,1.9375,2.0,
     *   2.0625,2.125,2.1875,2.25,2.3125,2.375,2.4375,2.5,2.5625,2.625,
     6   2.6875,2.75,2.8125,2.8750,2.9375,3.0,3.0625,3.125,3.1875,
     *   3.2500,3.3125 ,3.375,3.4375,3.5,3.5625,3.625,3.6875,3.75 /
      DATA XA,XE,HMIN,HMAX,EPS /0.0,20.0,.025,.1,.15/
      DATA VELIN /19.62/
C
C * * * * * * * * READ IN AND WRITE OUT KUTMER PARAMETERS AND PROGRAM
C       OPTIONS
      READ(5,HSP)
      WRITE(6,HSP)
      DO 10 I=1,6
   10 EPSE(I) = EPS
C
C * * * * * * * * SET UP CONSTANTS
      PI = 3.14159265359
      GRAVTY=32.18
      DPR=57.2957795130
      RPD=.0174532925199
      IF (EST(NUM).LT.BL) STOP 3
C
C          COMPUTE NO AND BM ARRAYS
C        THIS IS FOR SPECIAL BOW FORM ONLY. CHANGE PROGRAM
C       THRU STATEMENT 32 FOR NEW BOW SHAPE
C
      DO 32 I=1,NUM
      IF(EST(I).GE.0.75) GO TO 30
      NO(I)=-0.46875*(1.0-SQRT(EST(I)/0.375-(EST(I)/0.75)**2.0))
      BM(I)=.375*SQRT(1.0-(EST(I)/.75-1.)**2.0)
      GO TO 32
   30 NO(I)=0.0
      BM(I) = 0.375
   32 CONTINUE
C*******COMPUTE CONSTANTS AND INITIALIZE ARRAYS
      M=W/GRAVTY
      RHO=1.99
      IT=M*RG*RG
      PHALF = (PI/2.)*RHO
C
      BETA = BETA*RPD
      CD = COS(BETA)
```

```
      TA = TAN(BETA)
      DO 60  I=1,NUM
      E(I) = ECG-EST(I)
      N(I) = NCG-NO(I)
      HMAX(I) = KAR*PHALF*BM(I)*BM(I)
      TEST(I) = (2.*HM(I)*TA)/PI
   60 CONTINUE
      END=END+1
      RETURN
      END
      SUBROUTINE PLOTER (FX,XA,HMAX,IB,IPT)
C
C   INPUT:
C       FX        A TWO DIMENSIONAL ARRAY CONTAINING PITCH AND
C                 HEAVE VALUES AT EACH TIME STEP
C       XA        INITIAL TIME
C       HMAX      TIME INTERVAL, PTIME*HMAX = INTERVAL BETWEEN
C                 FX VALUES
C       IB        NUMBER OF FX VALUES
C
C
      REAL IT,K,LAMBDA,M,MA,MMAX,N,NCG
      INTEGER END
C
      DIMENSION FX(2,2000),FMIN(2),FMAX(2),NVAR(2)
C
      COMMON /CONST/ NCG,ECG,PI,DPR,RPD,GRAVTY,RHO,NUM,MA(120),CD,TA,
     *              B(120),BETA,MW(120),TZ,DRAG,W,XD,T,XP,M,IT,
     1           DELTAS,TX,EST(120),KAR,HMAX(120),TEST(120),
     *              N(120),PHALF
      COMMON/OUT/NPRINT,NPLOT,END
C
C * * * * * * * *  SET UP VALUES FOR PLOT AND CREATE PLOT
      NFUN=2
C * * * * * * * * SET UP MIN AND MAX LIMITS FOR PLOT
      FMIN(1)=FX(1,1)
      FMIN(2)=FX(2,1)
      FMAX(1)=FX(1,1)
      FMAX(2)=FX(2,1)
C
      DO 200 I=1,IB
      IF(FX(1,I).LT.FMIN(1))FMIN(1)=FX(1,I)
      IF(FX(1,I).GT.FMAX(1))FMAX(1)=FX(1,I)
      IF(FX(2,I).LT.FMIN(2))FMIN(2)=FX(2,I)
      IF(FX(2,I).GT.FMAX(2))FMAX(2)=FX(2,I)
  200 CONTINUE
C
  800 CONTINUE
      NVAR(1)=10H HEAVE
      NVAR(2)=10H PITCH
      N1=2
      X0=XA
      DELX = HMAX
      IF(NPLOT.EQ.1)CALL PLOT2(FX,FMIN,FMAX,NVAR,NFUN,N1,IB,X0,DELX)
      RETURN
      END
      SUBROUTINE TRAP(F,DX,NPTS,ANS)
C
C   INPUT:
```

```
C        F          ARRAY OF FUNCTIONAL VALUES OF THE INTEGRAND
C        DX         THE X INTERVAL BETWEEN VALUES
C        NPTS       THE NUMBER OF VALUES GIVEN
C   OUTPUT:
C        ANS        THE VALUE OF THE INTEGRAL
C
      DIMENSION F(NPTS)
      ANS=0.0
      IF(NPTS.LT.2)GO TO 999
      DO 1 I=1,NPTS
    1 ANS=ANS+F(I)
      ANS=DX*(ANS-0.5*(F(1)+F(NPTS)))
  999 CONTINUE
      RETURN
      END
      FUNCTION RMP(T,START,RISE)
C * * * * * * *  THIS FUNCTION IS USED TO GRADUALLY IMPLIMENT THE WAVE
C
C        T          CURRENT TIME
C        START      TIME TO START RAMP FROM 0.0 TO 1.0
C        RISE       THE LENGTH OF THE RISE FROM 0.0 TO 1.0
C
      H=0.0
      IF(T.LT.START)GO TO 99
      IF(RISE.EQ.0.0)GO TO 80
      TOP=T-START
      H=1.0
      IF(TOP.LT.RISE)H=TOP/RISE
      GO TO 99
   80 H=1.
      IF(T.EQ.START)H=0.5
   99 RMP=H
      RETURN
      END
      SUBROUTINE SEAWAY
      REAL K
      COMMON /WAVE2/ WO(20),K(20),C(20),RWO(20),RWO2(20),RK(20),
     & RO(20),RWK(20),H,PHS(20)
      COMMON /CONST/ NCG,ECG,PI,DPR,RPD,GRAVTY,RHO,NUM,MA(120),CD,TA,
     *        B(120),BETA,HW(120),TZ,DRAG,W,XD,T,XP,M,IT,
     1          DELTAS,TX,EST(120),KAR,MMAX(120),TEST(120),
     *          N(120),PHALF
      READ (5,80) H
      WRITE (6,90) H
   80 FORMAT (8F10.4)
   90 FORMAT (27H SIGNIFICANT WAVE HEIGHT = ,E10.4,2X,5H FEET,//)
      HROOT = SQRT(H)
      WN = 2.276/HROOT
      WO(1) = .7954*WN
      WO(2) = WN
      WO(3) = 1.183*WN
      WO(4) = 1.4036*WN
      WO(5) = 1.6015*WN
      WO(6) = 1.7953*WN
      WO(7) = 2.0035*WN
      WO(8) = 2.1941*WN
      WO(9) = 2.3919*WN
      WO(10) = 2.612*WN
      RO(1) = 0.1364*H
```

```
      RO(2) = 0.1861*H                                                  SEAWAY27
      RO(3) = 0.1657*H                                                  SEAWAY28
      RO(4) = 0.1302*H                                                  SEAWAY29
      RO(5) = 0.0999*H                                                  SEAWAY30
      RO(6) = 0.0771*H                                                  SEAWAY31
      RO(7) = 0.0604*H                                                  SEAWAY32
      RO(8) = 0.0482*H                                                  SEAWAY33
      RO(9) = 0.0390*H                                                  SEAWAY34
      RO(10) = 0.0626*H                                                 SEAWAY35
      PHS(1) = .005                                                     SEAWAY36
      PHS(2) = 2.41                                                     SEAWAY37
      PHS(3) = 5.20                                                     SEAWAY38
      PHS(4) = 4.00                                                     SEAWAY39
      PHS(5) = 0.00                                                     SEAWAY40
      PHS(6) = 1.27                                                     SEAWAY41
      PHS(7) = 3.11                                                     SEAWAY42
      PHS(8) = 2.92                                                     SEAWAY43
      PHS(9) = 3.55                                                     SEAWAY44
      PHS(10) = 0.70                                                    SEAWAY45
      DO 50 J=1,10                                                      SEAWAY46
      K(J) = WO(J)*WO(J)/GRAVTY                                         SEAWAY47
      C(J) = GRAVTY/WO(J)                                               SEAWAY48
      RWO(J) = RO(J)*WO(J)                                              SEAWAY49
      RWO2(J) = RWO(J)*WO(J)                                            SEAWAY50
      RWK(J) = RWO(J)*K(J)                                              SEAWAY51
      RK(J) = RO(J)*K(J)                                                SEAWAY52
   50 CONTINUE                                                          SEAWAY53
      RETURN                                                            SEAWAY54
      END                                                               SEAWAY55
      SUBROUTINE TABLE                                                  TABLE  2
      COMMON/SINE/POINT(1000)                                           TABLE  3
      DX = .003141592654                                                TABLE  4
      X = 1.570796327                                                   TABLE  5
      DO 100 J=1,501                                                    TABLE  6
      POINT(J) = SIN(X)                                                 TABLE  7
      K = 1002-J                                                        TABLE  8
      POINT(K) = -POINT(J)                                              TABLE  9
      X = X+DX                                                          TABLE 10
  100 CONTINUE                                                          TABLE 11
      RETURN                                                            TABLE 12
      END                                                               TABLE 13
```

DTNSRDC ISSUES THREE TYPES OF REPORTS

1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECHNICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.

2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIMINARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE. THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.

3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR INTERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE BASIS.